
HAPPY-COMPUTER
SPECTRUM SONDERHEFT
SONDERHEFT 1/1985
DM 14,-
HAPPY-COMPUTER
DAS GROSSE HEIMCOMPUTER-MAGAZIN
Kurs mit kompletten Beispiel-Listings:
Einstieg in Maschinensprache
Alles über die heißen Abenteuerspiele
Alle Sinclair-Schnittstellen unter der Lupe
Bastelanleitung
Soundgenerator und Schaltinterface
Über 20 neue Listings
★ Spiele ★ Anwendungen
★ Tips & Tricks
Viele Programme auch auf Kassette erhältlich
ZX Spectrum
REM PLOT-KORREKT.
PLOT OVER 1,XX,YY
RETURN
SAVE "PLATI.-CAD"

Mehr als ein Disksystem für den Spectrum

DISCOVERY 1

- neuestes 3½"-Laufwerk mit 180 KB
- Centronics-kompatibler Drucker I/O-Port
- Joystick Interface (Kempston-kompatibel)
- monochromer Video-Monitor-Anschluß
- stabilisiertes Netzteil (versorgt auch Ihren Spectrum)
- Befehlssatz voll Microdrive-kompatibel
- belegt **keinen** RAM im Spectrum
- deutsches Handbuch
- Gratis-Einführungsdiskette mit Programmen
- volle Herstellergarantie



DISCOVERY 1
DM 898.-*

DISCOVERY 2 (zwei Laufwerke) **DM 1398.-***, Druckerkabel **DM 59.-***, Disketten (10-Pack) **DM 99.-***

MHS Datenmanager Spectrum
anspruchsvolles Dateiprogrammsystem, Datensätze bis 5 KB, 1000 Datensätze im Indexzugriff, Deutsches Handbuch
(auf 3½"-Diskette) .... **DM 99,-***
(auf Microdrive Cartridge) ...... **DM 99,-***

MHS Tasword II Ergänzung
Ermöglicht zusammen mit Datenmanager Mail-Merging (Serienbriefe), Anpassung an jeden Adreßsatz.
(auf Kassette) .... **DM 29,90***
* unverbindliche Preisempfehlung inkl. MwSt.

Handbuch für Hacker
— Was wird zum Hacken benötigt? — Was ist eine Mailbox? (über 700 Nummern!!!) — Datex-P-BTX-TAB...
Für alle Rechner mit Modem; 2bändig, mit Aktualisierung .... **DM 38,-***

Die Discovery-Systeme gibt's ganz in Ihrer Nähe:

ELSCH COMPUTER
Leber Str. 11
1000 Berlin 62
Tel. 030/7826055



Raschplatz 9 h (Passerelle)
3000 Hannover 1 · Tel. (0511) 315411

AD FRIEDMANN
Informationssysteme
Wilhelminenstr. 17A
6100 Darmstadt
Tel. 06151/26566

LÜCKER MICRO-SOFT
In der Eisenbach 37
6270 Idstein
Tel. 06126/1559

CV COMPUTER
6450 Hanau 1, Steinheimer Str. 27
Telefon (0 61 81) 25 47 83



MICHAEL NAUJOKS
Rottmannstr. 40
6900 Heidelberg
Tel. 06221/46885

MHS Müller Hard & Software
Bergstraße 7
7262 Althengstett
Tel. 07051/3213

THORSTEN FREITAG
Soft & Hardwarevertrieb
8552 Höchstadt a. d. Aisch
Tel. 09193/7630

TRIEBNER & SCHWING Datensysteme

Denisstraße 45, Postfach 120330, 8500 Nürnberg, Tel.: 0911/552283



In oder out?

Alle Welt redet von 16 Bit, zählt RAM 100-Kilobyte-weise, träumt vom MOPS (Millionen Operationen pro Sekunde): Da ist der Spectrum »out«? Ich glaube nicht. Es gibt Fachleute, die behaupten, zwei Drittel bis drei Viertel aller Mikrocomputer-Anwender bräuchten ohnehin nicht mehr als die Leistung eines 8-Bit-Prozessors. Für den Spectrum spricht ferner, daß er preisgünstig und ausgereift ist, daß es sehr viel Hard- und Software gibt — und eine Menge Literatur und Erfahrungen. Seitdem die Plus-Version geliefert wird, kann man nicht einmal gegen das Tastaturformat etwas einwenden. Selbst manche systembedingten Beschränkungen sind bei näherer Betrachtung gar nicht so wichtig: So reicht eine Bildschirmdarstellung von maximal 64 Zeichen/Zeile auch für Textverarbeitung völlig aus (wer's nicht glaubt, sollte einmal die durchschnittlichen Anschlagzahlen in einem mit Schreibmaschine geschriebenen Brief zählen).

Natürlich ist ein Sechszylinder-Mercedes schöner und komfortabler und schneller als ein R4. Ob die Anschaffung aber einen Sinn macht, wenn Sie mit dem Auto im wesentlichen Ihre Briefe zur Post bringen und alltägliche Besorgungen erledigen wollen, ist eine ganz andere Frage. Das gleiche gilt für den Computer-Kauf: Es muß ein vernünftiges Kosten/Nutzenverhältnis gegeben sein. Und da sieht der Spectrum wirklich nicht schlecht aus. Deswegen ist er immer noch »in« — selbst und gerade bei den vielen Benutzern, die ihn gut kennen.

(Michael Pauly, Chefredakteur)

Sinclair for ever

Angespornt durch viele nette Zuschriften mit oben genanntem Ausspruch habe ich mich an unser zweites Sonderheft gewagt. Geplant war, in diesem auch als Aufsteiger-Alternative über den QL in der deutschen Version zu berichten. Da es zu diesem Computer aber bisher trotz vieler Versprechungen nur Absichtserklärungen und allenfalls Gerüchte gibt und die deutsche Niederlassung von Sinclair in Bad Homburg nicht gerade durch positive Aktivitäten von sich reden macht, haben wir auf den QL verzichten müssen. Rüstet man den Spectrum mit einem Diskettensystem aus, hat man auch wesentlich preiswerter eine »Profi-EDV-Anlage«. Hier ist zum Beispiel das Opus-Discovery-System eine ernsthafte Betrachtung wert. Dafür gibt es inzwischen den Spectrum plus und für Leute, die schon einen Spectrum haben, einen Aufrüstsatz für knapp 100 Mark. Dieser besteht aus der Plus-Tastatur, der Demo-Kassette und dem neuen Handbuch und macht unter Verwendung eines Schraubendrehers aus jedem Spectrum einen Spectrum plus. Ebenfalls erfreulich ist die Tatsache, daß das Interface 1 mit Microdrive zu einem günstigen Preis offeriert wird und nun wohl endlich damit gerechnet werden darf, daß auch eine Vielzahl kommerzieller Programme auf Cartridge angeboten wird. Die extrem stark gesunkenen Preise des Microdrive-Cartridge (das Viererpack schon für 32 Mark) trösten zusätzlich über die Anschaffungskosten für das Interface mit Laufwerk hinweg. Ein Diskettenlaufwerk kostet immer noch rund das Doppelte.

Was geboten wird

Die Themenauswahl zu diesem Sonderheft haben Sie vorgenommen, mit Ihren Leserkarten, Briefen, Anrufen, Programm- und Artikelangeboten. Wir gehen davon aus, daß nicht jeder Spectrum-Besitzer ein alter Hase ist. Darum sind einige Tips und Tricks und Spielelistings für Einsteiger gedacht. Aber auch schlaue Füchse sollen zu ihrem Recht kommen. Diesen sei der Maschinencode-Kurs ebenso wie der Artikel über die Spectrum-Schnittstellen ans Herz gelegt. Mit Testberichten haben wir uns zurückgehalten, sie sind Bestandteil der aktuellen monatlichen Ausgabe der Happy-Computer. Damit die Lötkolben-Bändiger was zum »Verbraten« finden, haben wir einen alten Bekannten, unseren Soundgenerator, nochmal hervorgeholt und dafür auch gleich und exclusiv eine 8-Bit-Ausgabeeinheit zum Selberstricken im Angebot. Für Musestunden oder als Entspannung zwischen der Abtipperei bieten sich der Adventure-Artikel und die Psion-Story an. Zum Nacheifern sei die »goldene Diskette« empfohlen, nicht zum Nachmachen (oder doch?) soll die »Spectrum-Reparatur« anleiten.

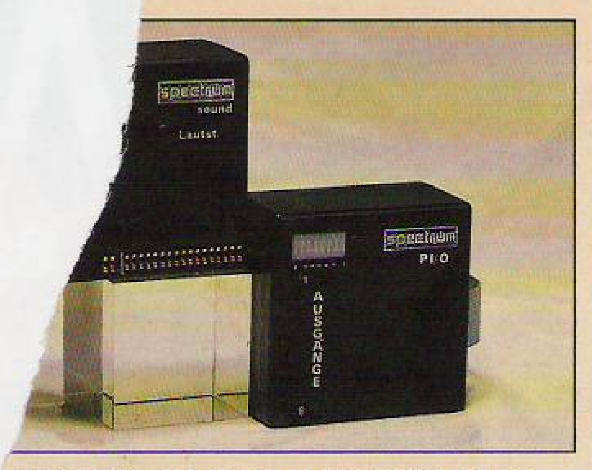


Eine Menge Listings

Die anwendungsbezogenen Listings sollen dem Gebrauch als auch, wie natürlich speziell die Tips und Tricks-Listings, dem Denkanstoß für eigene Programmentwicklungen dienen. Viele Programme aus dem Listing-Teil bieten wir, um Ihnen das lästige zeitraubende und nervenzehrende Eintippen zu ersparen, auf einer Sammelkassette (Bestellnummer LH85S1D, 19,90 Mark) an. Der Preis dafür ist nur noch durch Raubkopieren zu unterbieten.

In der Hoffnung, allen Spectrum-Fans einen Gefallen getan zu haben, Ihr Happy Computer-Redakteur und überzeugter Spectrum-Benutzer

(Manfred-D. Kotting)



auvorschläge machen aus dem müden Beep
n Sound und einen Ausgabe-Baustein mit

```
*HISOFT GENS3M2 ASSEMBLER*
      ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

EA60              1        ORG  60000
               2091 *L+
               2092 ***
               2093 *** LISTING 8
               2094 ***
EEE5 3620      2100 TEILOE LD   (HL)," "
EEE7 23        2110        INC  HL
EEE8 10FB      2120        DJNZ TEILOE
EEEA C9        2130        RET

Pass 2 errors: 00

Table used:    460  from   491
```

Ein Maschinencode-Kurs für Fortgeschrittene mit einer kompletten Unterprogramm-Sammlung und leicht verdaulicher Anleitung hilft auch Ihnen beim Programmieren.



Daß Psion mehr kann, als die Demo-Kassette Ihres Spectrum zu produzieren, zeigt unser Besuch in England.

Spiele-Listings

Tips & Tricks-Listings

Anwendungs-Listings



Für Elektronik-Bastler bieten wir ein Platinen-CAD-Programm, das ihnen die Arbeit wesentlich erleichtert.



Den Signalen des Spectrum geht unser Artikel »Verbindungen« auf den Grund. Damit können Sie mehr aus Ihren Daten machen.



Wenn Sie sich in ein Abenteuer mit dem Spectrum stürzen wollen, finden Sie in unserem Adventure-Artikel viele Tips und Anregungen.

Abenteuer Spectrum

Die Schuhe geschnürt, den Ranzen gepackt und auf geht's: Ihr Computer entführt Sie in verwunschene Schlösser, unheimliche Höhlenlabyrinthe und ferne Welten. Adventures machen es möglich, die anspruchsvolle Spielart, bei der nicht Reaktionen und Schnelligkeit, sondern Überlegung und Gehirnschmalz gefragt sind. In diesem Abenteuerspiel-Führer stellen wir Ihnen die interessantesten Programme dieses Genres vor.

Waren Computerspiele in der Regel Programme, bei denen man Punkte sammelte und für die man sich ein Joystick-Interface zulegte, eroberte ein neuer Spieltyp klammheimlich eine stolze Fan-Gemeinde. Selbst abgeklärte, sachliche Menschen, die mit Spielereien sonst wenig im Sinne haben, hocken bis tief in die Nacht vor der Glotze und grübeln im fahlen Licht einer einsamen Schreibtischlampe vor sich hin. Verursacher dieser harmlosen nächtlichen Leidenschaft sind der gute alte Spectrum und ein entsprechendes Programm: Abenteuerspiel oder neudeutsch Adventure genannt.

Bei Adventures gilt es immer eine bestimmte Aufgabe zu lösen, egal ob ein böser Zauberer unschädlich gemacht, ein Schatz gefunden oder ein Mordfall aufgedeckt werden muß. Um diese Aufgabe zu erfüllen, muß man sich in einer neuen Welt, die nun sozusagen im Spectrum steckt, zunächst orientieren. Gute Abenteuerspiele enthalten nämlich eine Vielzahl von Schauplätzen, Gegenständen und auch mehr oder weniger freundlich gesonnenen Personen. Einfach wird die Sache dadurch nicht, aber ein Abenteuerspiel, das schon nach kurzer Zeit gelöst wäre, ist auch nicht das Gelbe vom Ei. Einmal zu Ende gespielt, wird ein Adventure nämlich reizlos, die Spielmotivation geht völlig flöten.

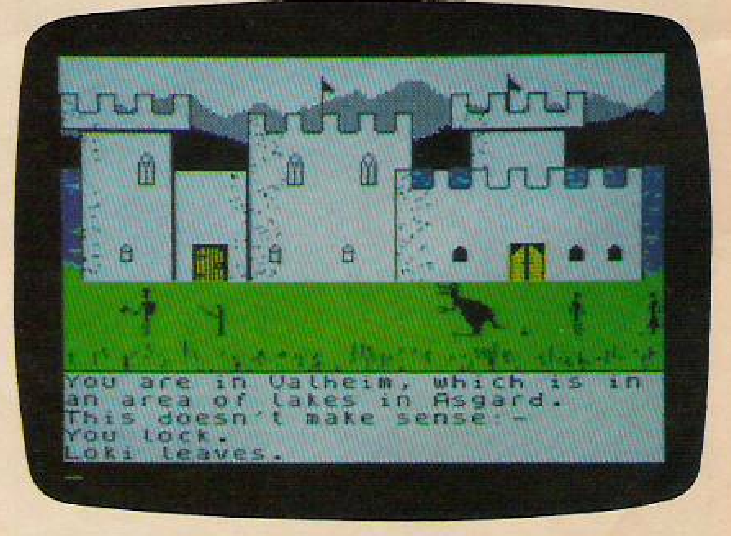
Die Welt im Computer

Angenommen, das Programm teilt Ihnen am Bildschirm nach dem Laden mit, daß Sie auf einer Waldlichtung stehen. Mehr tut sich aber auch nicht, denn der Spectrum wartet nun darauf, daß Sie die Initiative ergreifen. Um nun beispielsweise nach Norden zu gehen, tippt man im schönsten Schulenglisch »Go North« ein. Nun tut sich auf dem Bildschirm wieder etwas: Sie erreichen einen neuen Schauplatz, an dem Sie sich wieder orientieren müssen. Nach diesem Schema läuft das ganze Adventure ab. Durch Eingeben Ihrer Befehle über die Tastatur verraten

Der nette Herr links scheint auf den ersten Blick ein Verkäufer in einem Computershop zu sein — doch weit gefehlt: der »Ringmaster« ist's, frisch aus dem Adventure »Spider-Man«



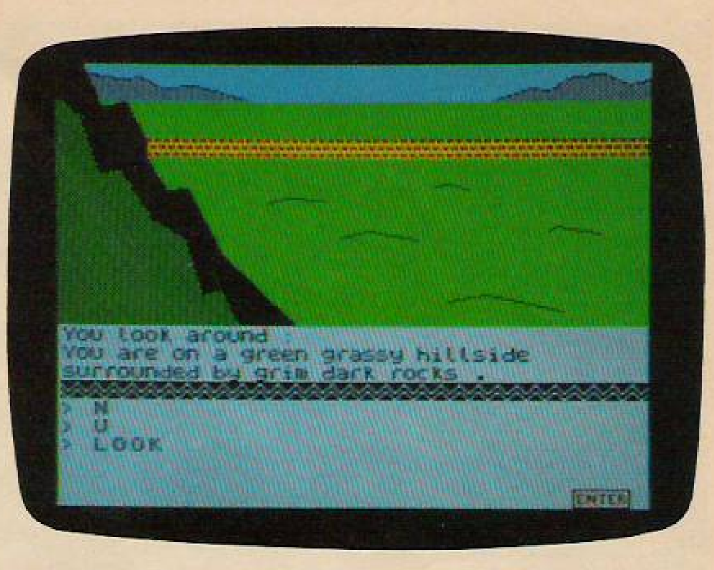
Man plätschert sich durch: »Forest at Worlds End«



Götter, Grusel und Gefahren: »Valhalla«



Frisch aus dem Dom entkommen: »The Hulk«



Unterwelt mit Grünstreifen: »Inferno«

Sie dem Programm, was Sie als nächstes in der Phantasiewelt tun wollen. Da nahezu alle Adventures aus England kommen, sollten Sie sich auf alle Fälle ein Wörterbuch zurecht legen, denn deutsche Befehle versteht ein englisches Programm nicht. Es gibt zwar einige deutschsprachige Spectrum-Abenteuer, die aber bei weitem nicht die Qualität der besten englischen Programme erreichen.

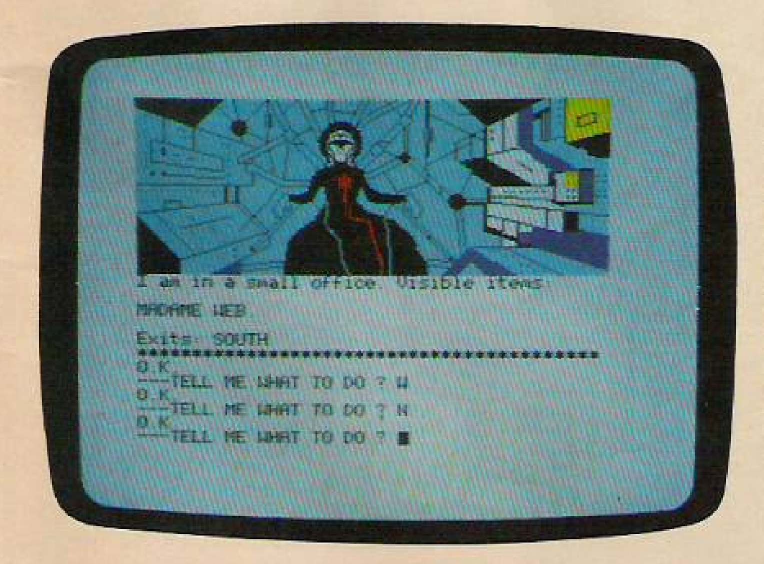
Grafik kontra Text

Abenteuerspiele kann man im großen und ganzen in zwei wesentliche Kategorien einteilen: Zum einen mit Grafik, zum anderen Text pur. Reine Textadventures sind optisch natürlich recht unattraktiv, worunter der Spielwitz spürbar leidet. Doch bei vielen Programmen kredenzt der Spectrum mehr oder weniger schöne Abenteuerbilder, wobei einige Programmierer erstaunlich viel Text und erstklassige Grafik in den 48 KByte-Speicher packen. Vor allem dem Einsteiger sind Grafikadventures zu empfehlen, da sie mehr Spaß machen und auch nicht wesentlich teurer sind als knochentrockene Textorgien.

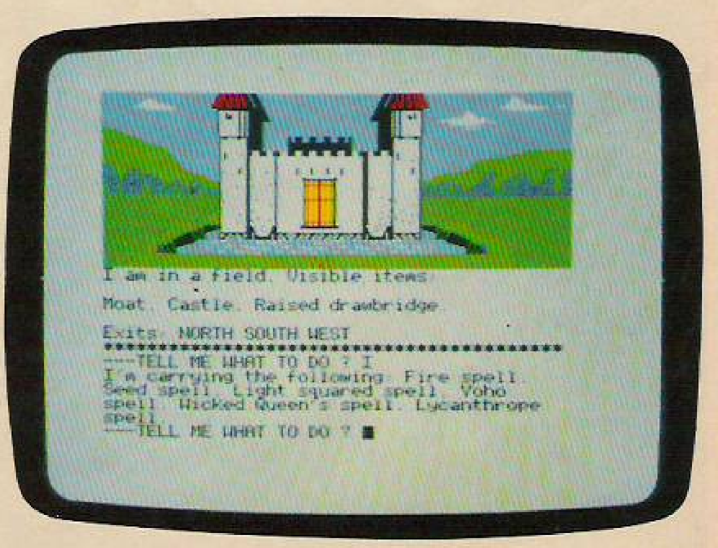
Bei besonders anspruchsvollen Texten nimmt man auch gerne mal eine relativ mickrige Grafik in Kauf, so zum Beispiel bei »Sherlock Holmes«. Das Programm steht dem großen literarischen Vorbild kaum nach, wenn es um detaillierte Beschreibungen und verzwickte Handlungsfäden geht. Sherlock Holmes und Dr. Watson müssen natürlich einen Mordfall in London klären; eine schwierige Aufgabe, vor der Anfänger gewarnt seien. Krimifans und Freunde sprachlich anspruchsvoller Abenteuerspiele werden bei »Sherlock Holmes« schier aus dem Häuschen sein, so viel Atmosphäre kommt bei diesem Computer-Mordfall rüber.

Comic-Strip als Adventure-Hit

Eine Art »Sherlock Holmes der Adventure-Programmierer« ist der Engländer Scott Adams, der so viele professionelle Abenteuerspiele geschrieben hat wie sonst niemand auf der Welt. Von ihm stammen die beiden Comic-Titel »The Hulk« und »Spider-Man«. Vor allem jugendliche Fans werden an den fantasievollen Programmen viel Spaß haben. Ein weiterer Scott Adams-Titel dreht sich um den bösen Herrscher eines verwunschenen Schlosses: »The Sorcerer of Claymorgue Castle« ist ein stimmungsvolles Abenteuer, bei dem viel Magie mit im Spiel ist. Alle



Jede Menge komische Typen: »Spider-Man«



Ein schickes Schlößchen: »Sorcerer of Claymorgue Castle«

Titel	Preis	Grafik	Softwarehaus	Besonderheiten
Adventure Quest	39,–	Nein	Level 9	Eigenständige Fortsetzung zu »Colossal Adventure«. Text pur mit 225 Schauplätzen.
City of Ehdollah	39,–	Nein	Goldstar	Ruinengestöber mit allerlei Monstern. Ein geheimnisvoller Rubin soll gefunden werden.
Dungeon Adventure	39,–	Nein	Level 9	Eigenständige Fortsetzung zu »Adventure Quest«. Höhlenwanderung ohne Grafik.
Colossal Adventure	39,–	Nein	Level 9	Betagter Klassiker von 1983. Eine Höhlenerforschung für Nostalgie-Fans.
Doomdark's Revenge	39,–	Ja	Beyond	Eigenständige Fortsetzung zu »Lords of Midnight«. Noch mehr strategische Elemente.
Emerald Isle	32,–	Ja	Level 9	Als Pilot machen Sie Bruchlandung auf einer Insel. Über 200 Schauplätze.
Forest at Worlds End	29,–	Ja	Interceptor	Eine Prinzessin soll gerettet werden. Zur edlen Handlung gibt's einige schöne Grafiken.
Hampstead	39,–	Nein	Melbourne House	Originelle Handlung: Ziel des Spiels ist der soziale Aufstieg
Inferno	33,–	Ja	Richard Shepherd	Witziges Unterwelt-Abenteuer mit Zerebus, Minos & Co.
Lords of Midnight	39,–	Ja	Beyond	Strategische Elemente, Tastaturschablone mit den Befehlen liegt bei, sehr komplex
Lords of Time	39,–	Nein	Level 9	Reise durch diverse Zeitepochen. Keine Grafik, aber über 200 Schauplätze
Return to Eden	39,–	Ja	Level 9	Science fiction-Abenteuer im 24. Jahrhundert mit über 200 Bildern.
Sorcerer of Claymorgue Castle	39,–	Ja	Adventure International	Zaubersprüche müssen im richtigen Moment angewendet werden. Sehr schöne Bilder.
Sherlock Holmes	49,–	Ja	Melbourne House	Schwierig mit anspruchsvollen Texten. Gute Englischkenntnisse empfohlen.
Spider-Man	39,–	Ja	Adventure International	Für Comic-Fans: Der Abenteuerer schlüpft in die Rolle von »Spider-Man«. Schöne Grafik.
The Hobbit	59,–	Ja	Melbourne House	Klassiker in Anlehnung an Tolkiens »Herr der Ringe«. Handlung und Grafik vom feinsten.
The Hulk	49,–	Ja	Adventure International	Grafisch überzeugendes Spiel um die Comic-Figur »Hull«. Nicht gerade einfach.
Valhalla	49,–	Ja	Legend	Die Figuren agieren auf dem Bildschirm wie in einem Zeichentrickfilm, sehr reizvoll.

Bezugsquelle für alle Programme: Joysoft, Humboldtstr. 84, 4000 Düsseldorf 1, Tel. (02 11) 680 14 03

Eine Auswahl von Spectrum-Adventures

drei Titel haben eines gemeinsam: Die zahlreichen Grafiken gehören mit zum Besten, was man in dieser Kategorie bisher auf dem Spectrum gesehen hat. Prädikat: sehr empfehlenswert.

Ein Programm, das ziemlich aus dem sonst üblichen Adventure-Rahmen fällt, ist das Götterdrama »Valhalla«. In diesem Spiel erscheinen keine starren Bilder, sondern es tut sich wie beim einem Zeichentrickfilm eine ganze Menge. Verschiedene Figuren tauchen auf, tauschen Gegenstände, bekämpfen sich und verschwinden wieder. Der Spieler ist natürlich auch mit einer Figur vorhanden und kann, wie sich das für ein Abenteuerspiel gehört, Kommandos über die Tastatur eingeben. Kreuzt beispielsweise Odin auf und Ihnen ist nach einer kleinen Rauferei zumute, geben Sie »Kill Odin« ein. Nach einigen Sekunden Verarbeitungszeit tippelt Ihre Spielfigur auf dem Bildschirm prompt zu Odin hin und ein Handgemenge nimmt seinen Lauf. Wie gesagt: Kein Adventure im herkömmlichen Sinn, eher ein einfallsreiches Trickfilmabenteuer. Kein Wunder, daß das originelle Programm 1984 in England alle möglichen Preise gewann. Auch das Handbuch zu »Valhalla« bestätigt den sehr guten Eindruck. Vor allem die Zeichnungen der Sagenfiguren sind ein Augenschmaus.

Kino-Erfolge werden umgesetzt

Kurz nach Redaktionsschluß und deshalb ohne Bildschirmfoto kam »Gremlins« auf unseren Schreibtisch. Es ist das Adventure zum letztjährigen Steven Spielberg-Film und wurde von den Routiniers von Adventure International (»The Hulk« etc.) in gewohnter Qualität programmiert. Die Grafik ist sehr gut, doch das Spiel selbst ziemlich schwierig, vor allem, wenn man den Film nicht gesehen hat. Anfänger seien gewarnt.

Soweit einige Besonderheiten aus dem Angebot an Spectrum-Adventures. Eine ausführliche Übersicht, die keinen Anspruch auf Vollständigkeit erhebt, haben wir für Sie in einem Kasten zusammengefaßt. Und falls Sie jetzt so richtig auf den Geschmack gekommen sind, noch etwas indiskrete Eigenwerbung: Die Tests von allen guten Neuerscheinungen finden Sie jeden Monat in »Happy Computer«. Happy Adventureing...

(Heinrich Lenhardt)

Klavierstunden für 27 Mark

Ein gutes Musik-Programm macht aus dem Spectrum einen geduldigen Musiklehrer.

Mehr und mehr »Videospielmüde« entlocken in letzter Zeit ihren Heimcomputern teils abenteuerliche Klänge und Tonfolgen: Musik ist drin. Musicmaster greift all den Spectrum-Besitzern unter die Arme beziehungsweise Finger, die sich mangels musikalischer Vorbildung bisher nicht an diesen kreativen Zeitvertreib heranwagten oder immer noch der Auffassung sind, man könne mit einem Piepser keine Musik machen. Klar, auch die beste Software wird den Spectrum nie in einen Konzertflügel verwandeln. Doch warum sofort nach dem Höchsten streben? Auch ein Hirte erzeugt mit seiner primitiven Pfeife Töne, Melodien, Musik und hat viel Spaß dabei.

Musicmaster ist gedacht für absolute Musiklaien. Die Software lüftet in Bild und Ton die Geheimnisse um Tonhöhen, Zeitwerte, Pausen, Vorzeichen und Taktarten. Die Voraussetzung für das erfolgreiche Bestehen des Software-Mini-Musikkurs liegt in ganz anderem Bereich. Denn, nichts geht ohne Englisch-Kenntnisse. Alle Erklärungen im »Bildschirmtext« sind in Englisch abgefaßt. Doch welcher wahre Computerfreak ist dieser Sprache neben Basic und Assembler nicht fähig? Und wenn, dann ist es höchste Zeit, sich einen Software-Vokabeltrainer zuzulegen.

Hat man die Lektionen des Musicmaster durchgearbeitet, weiß man etwas mehr als zuvor. Vom Musikprofessor ist man jedoch noch sehr weit entfernt. Und leider, oder Gott sei Dank, der Computer erklärt zwar, wie die Töne heißen und klingen, aber nicht wie man sie aneinanderreiht, damit das, was herauskommt, auch Musik ist. Hierzu ist letztlich doch ein gewisses Quantum Musikalität nötig. Die wahren Musiker beweisen sich also erst in Teil 2 der Musicmaster-Software.

Im Stave- oder Keyboard-Mode funktioniert das Programm den Spectrum zum Miniklavier beziehungsweise zur Automatikspieluhr um. Hier werden Töne zu Melodien. Akkorde lassen sich jedoch nicht spielen, sondern immer nur monophone Solos. Der Piepser des Spectrum kann nicht mehrere Töne gleichzeitig produzieren.

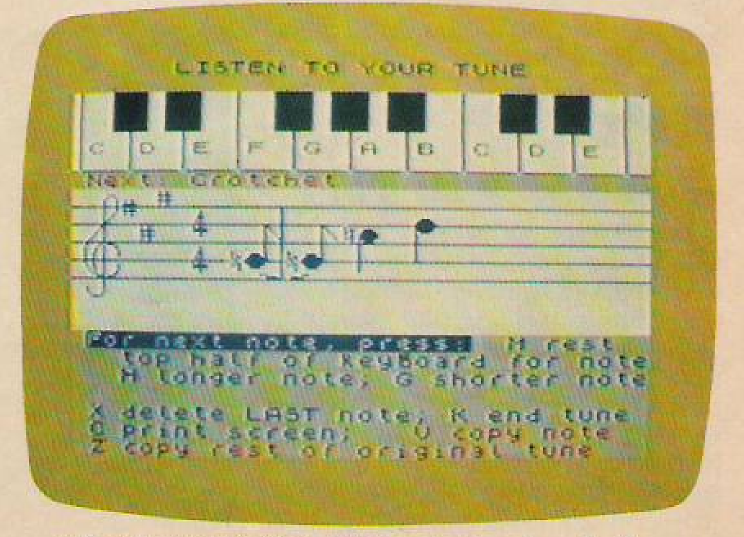
Die Töne gibt man durch Drükken der zugehörigen Symbole, also C, D, E und so weiter und einem zusätzlichen Hilfszeichen für erhöhen, erniedrigen beziehungsweise normal ein. Oder man legt das mitgelieferte Overlay über die alphanumerische Tastatur des Spectrum. Die beiden oberen Tastenreihen bilden dann eine Miniklaviatur mit schwarzen und weißen Tasten. Nun spielt man Ton für Ton genau wie auf einem Klavier. In jedem Fall erscheint der eingegebene Ton sofort auf der richtigen Notenlinie am Bildschirm. Hat man in Lektion 1 der Software gut aufgepaßt, fällt es nun nicht schwer, den richtigen Ton zu treffen und mit den speziellen Befehlen Taktart, Tempo, Tonhöhe, Tonlänge, Vorzeichen und Pausenwerte den Vorstellungen entsprechend einzugeben. Natürlich kann man falsche Eingaben später korrigieren, ganze Abschnitte löschen oder kopieren.

Wie sehr man sich selbst als Musik-Maestro bewährt, hört man spätestens beim Abspielen der eingegebenen Melodie. Und braust Beifall auf, sollte man spätestens jetzt das gelungene Werk auf Kassette speichern. (Richard Aicher)



Das Menü läßt den Programmumfang erahnen ►





Mit Musicmaster können Sie lernen ...

... und auch ausgiebig üben, Noten zu lesen und zu schreiben

Britischer Humor

Sinclair hat es schwer, nicht nur mit seiner deutschen Repräsentanz, sondern auch mit den Finanzen und der deutschen QL-Version. Aber es gibt auch einen Lichtblick: den Spectrum plus.

Vor gut einem Jahr startete Sinclair einen Werbefeldzug für den QL in deutscher Überarbeitung. Gleichzeitig übernahm eine neue Mannschaft die bisher vom Generalvertreter Schumpich geleistete Arbeit, es wurde eine Niederlassung in Bad Homburg etabliert. Im Herbst 1984 kam dann überraschend der Spectrum plus und ebenso überraschend ging das Homburger Führungstrio. Zum eingedeutschten QL gibt es bisher nur Absichtserklärungen. Daß zwischen der Präsentation des C5 (kein Computer, sondern ein Plastik-Elektro-Dreirad) und dem abgebrochenen Versuch der Einführung einer Sinclair-Aktie an der britischen Börse ein Zusammenhang besteht, ist nicht zu beweisen.

Auch das Weihnachtsgeschäft lief für Sinclair weder in England noch in Deutschland zufriedenstellend. Nur der Spectrum plus erfreute sich anfangs großer Beliebtheit, die Nachfrage war größer als das Angebot. Der QL in der britischen Original-Version wird über mehr oder weniger offizielle Importwege nach Deutschland gebracht und seit einigen Wochen ist dieser Computer nun auch offiziell bei Sinclair in Deutschland erhältlich. Von den bisherigen »grauen« Importen unterscheidet er sich nur in der mitgelieferten Psion-Software. Diese wird in der Version 2.00, einer wesentlich verbesserten Version, ausgeliefert.

Was es neues gibt

Der Spectrum ist offenbar auch aus der Sicht der Zubehör-Industrie ausgereizt. Es gibt seit Monaten keine Peripherie-Hits. Dafür ist wenigstens das Preisniveau auf eine vernünftige Ebene gesunken, und Sinclair forciert endlich etwas mehr sein Microdrive. Das Set aus Interface 1 und Laufwerk mit vier Cartridges und vier Programmen ist preiswert zu bekommen und wird hoffentlich das Angebot an Software für dieses Speichermedium erhöhen. Seit einigen Wochen munkelt man in England über ein weiteres Mitglied in der Spectrum-Familie, eine tragbare Version mit integriertem Laufwerk, LCD-Anzeige und Akku-Puffer. Wenn dies stimmt und dieser Neue auch noch kompatibel zu seinen Brüdern ist, könnte er eine Chance haben. Auf dem QL-Sektor ist es in bezug auf Peripherie bedauerlich ruhig und in bezug auf Software gefährlich leise. Dies gilt eingeschränkt für den britischen und uneingeschränkt für den hiesigen Markt. Selbst auf der LET, Englands großer Computer-Zubehör-Messe, war im Februar das Thema QL nicht sehr gefragt.

Nur keine Panik

Nach neuesten Meldungen fehlen Sir Clive nur 15 Millionen britische Pfund an Liquidität, um seine Verbindlichkeiten und Neuentwicklungen zu finanzieren. Grund dieses Finanzloches ist der erwartungsgemäß schleppende Absatz des Dreirades (C5), des Taschenfernsehers (drei Jahre zu spät) und des QL (Pseudo-Personal-Computer). In der entsprechenden Presseverlautbarung liest sich dies dann so:

1. Das Unternehmen Sinclair Research Limited bestätigt, daß es Kapital in Höhe von 10 bis 15 Millionen Pfund in der Industrie oder anderweitig aufnehmen will, um langfristiges Wachstum und Restrukturierungspläne zu finanzieren.

2. Sir Clive Sinclair sucht seit einiger Zeit für das Unternehmen einen neuen Chief Executive Officer (CEO) für eine Position, die er bisher selber innehatte. Diese Maßnahme ist Teil einer Firmenrestrukturierung, die mit der Schaffung von zwei Betriebsdivisionen von Sinclair Research im März begonnen worden ist. Die Computer Division sowie die TV & Communications Division haben beide je einen eigenen Managing Director.

Die sicherlich richtige Entscheidung, nun endlich einen verantwortlichen und kaufmännisch erfahrenen Mann in das Unternehmen zu holen, kommt hoffentlich nicht zu spät. Spitze wäre, wenn dies auch für die deutsche Dependence in Bad Homburg gelingen würde, die noch immer steuerlos im Abwind driftet.

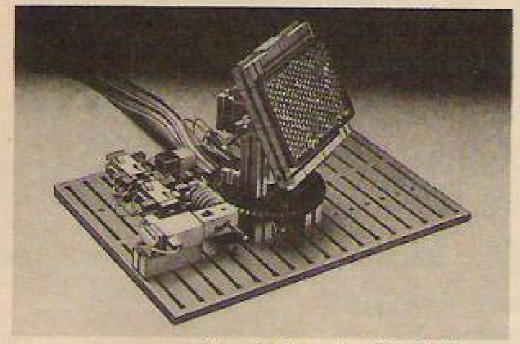
(Manfred Kotting)

Wo bitte geht's zum Markt?

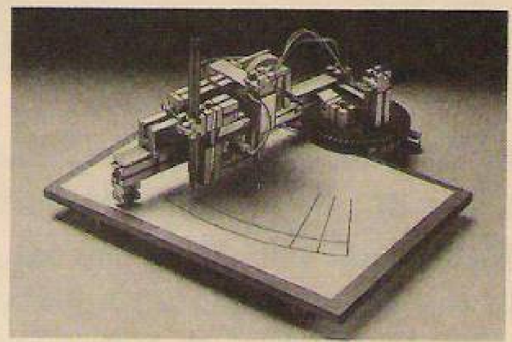
Sowohl IBM und Atari als auch Commodore haben neben Sir Clive Sinclair eine Marktlücke zwischen den bisherigen Gruppen der Heim- und Personal Computer entdeckt. IBM machte mit seinem PC jr. nach zwei Fehlstarten eine Bruchlandung. Mit seinen CPCs drängt nun »von unten« Schneider in den gleichen Anwender-Bereich. Aber der Z80-Computer wird gegen die 16-Bitter in diesem Marktsegment, schon wegen der Prozessor-Gläubigkeit (16 Bit ist in), keine Chance haben. Ob überhaupt in der Lücke zwischen dem Heimcomputer bis zu 1000 Mark und dem Personal Computer ab 5000 Mark aufwärts ein nennenswerter Markt vorhanden ist, ist ungewiß. Ob dort Platz für den QL ist, wird von mir bezweifelt. Eventuell tummeln sich in der Preisklasse des QL nur ein paar Technologie-Freaks aus der Heimcomputer-Szene und einige finanzschwache Personal-Einsteiger. Sicherlich negativ hat sich die lange Wartezeit auf den QL in deutscher Version für Sinclair ausgewirkt. Die oben erwähnten Freaks haben sich mit der preiswerteren britischen Version bedient, die ja bis auf die Sprache voll kompatibel sein soll. Nur die von Sinclair angepeilten Abnehmer (Gewerbetreibende, Ärzte und andere Selbständige) legen Wert auf deutschsprachige Anleitung und Bedienerführung. Ein Verkaufshit dürfte der eingedeutschte QL also kaum werden.

(Manfred Kotting)

Wer seinen

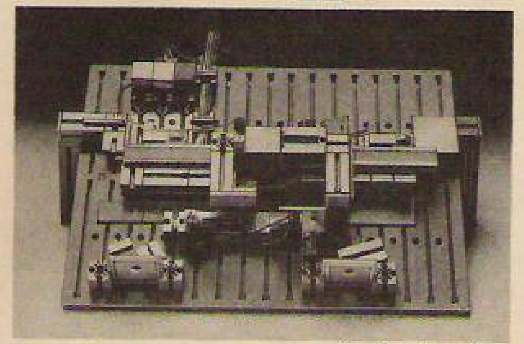


Eine Solarzellen-Nachführung

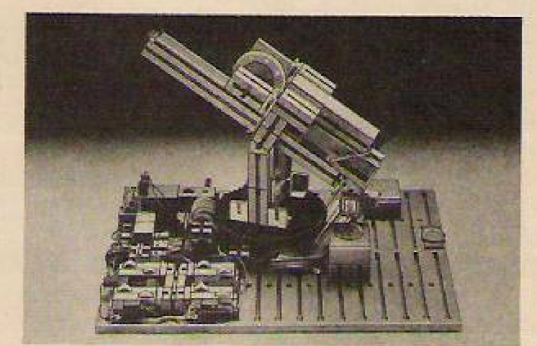


oder ein Plotter,

Computer liebt,

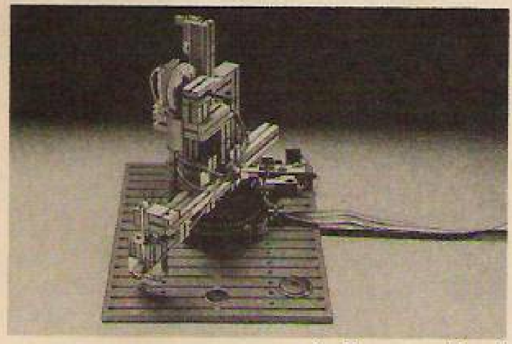


eine Sortieranlage

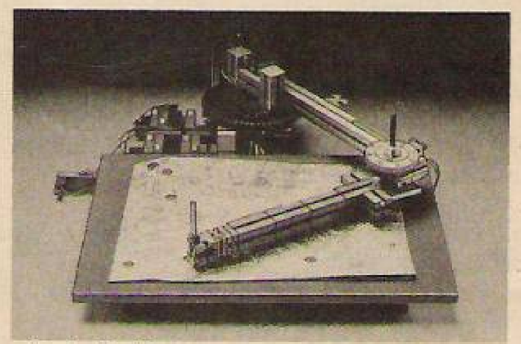


oder ein Teach-in Roboter,

schenkt ihm einen



der Turm von Hanoi



oder ein Grafiktablett – alles aus einem Baukasten.

Baukasten.

fischertechnik computing – das ist das neue System, mit dem man noch mehr Leben in den Home-Computer bringen kann. Aus einem einzigen Baukasten lassen sich 10 und mehr Peripheriegeräte konstruieren und programmieren. fischertechnik computing – über ein passendes Interface/Software-Paket kompatibel zu vielen gängigen Home-Computern.

Wir schicken Ihnen gern die komplette Info-Mappe und sagen Ihnen, wo Sie fischertechnik computing kaufen können. Einfach Coupon ausfüllen und einsenden an: fischer-werke, Weinhalde 14-18, D-7244 Tumlingen/Waldachtal.

Name

Straße

PLZ/Ort

fischertechnik®

Technik. Mit Zukunft.



BRAUCH, HUBE & PARTNER

Psion — eine Firma mit

Der Name Psion steht für Sinclair-Benutzer ganz oben, wenn es um hervorragende Software für ZX81, Spectrum und nun auch den QL geht. Tatsächlich hat die Firma einen kometenhaften Aufstieg hinter sich, und ein Ende ist noch nicht abzusehen. Wer und was steckt hinter diesem Softwarehaus? Dr. Bosetti war für uns in London und hat Psion unter die Lupe genommen.

Das Taxi hält in der Huntsworth Mews bei Psion Ltd., wo ich um 11 Uhr mit David Potter, dem Gründer und Chef der Firma verabredet bin. Eine leichte Verwunderung kommt auf, das Gebäude macht eher den Eindruck einer kleinen Bastlerwerkstatt denn den einer Software-Firma, die einen Umsatz von 10 Millionen Pfund im Jahr erreicht.

Im Hause selbst erfahre ich die Lösung: Das Hauptquartier von Psion ist verlegt worden, hier befindet sich praktisch nur noch das Versandlager. Zum Glück ist das neue Haus nur 100 Meter entfernt. Schon von außen kann man hier den Erfolg der letzten Jahre erkennen, ein dezentes goldenes Türschild und der Stil erinnern an Downingstreet 10.

Zwei junge Damen sitzen in der Empfangshalle. David Potter wird sofort hier sein, berichten sie mir. In den wenigen Minuten Wartezeit bekomme ich einen Eindruck von der Aktivität von Psion, das Telefon steht keine 10 Sekunden still. Die Damen haben wirklich einen »Full Time Job«.

Während ich noch an den Wänden die Produkte bewundern kann, die Psion groß gemacht haben — VU 3D, Hungry Horace, Flightsimulation und wie sie alle heißen — kommt auch schon David Potter, um mich abzuholen. Er ist es, der diese Firma als Ein-Mann-Betrieb Ende 1980 gegründet hatte. Nicht ohne Stolz erzählt er, daß er das Startkapital für Psion an der Börse »gemacht« hat, immerhin etwa 50000 Pfund.

Vielleicht hat man nun den Eindruck, es handele sich um einen reinen Wirtschaftsmanager, aber weit gefehlt. Ehe Potter die Firma gründete, war er als promovierter Physiker an den Universitäten in Los Angelos und London als Lehrer tätig, und er hat lange Zeit überlegt, ob er seine akademische Laufbahn aufgeben soll.

Betrachtet man die Gehälter an den britischen Universitäten, so kann man wohl sagen, daß sich zumindest in finanzieller Hinsicht sein Entschluß als richtig erwiesen hat. Zur Zeit beschäftigt seine Firma etwa 65 Mitarbeiter und ist weiter auf Expansionskurs.

Angefangen hat alles mit Spielprogrammen für den ZX81 und Psion hat damals gezeigt, was man alles aus diesem Computer herausholen kann. Noch heute sollen sich einige dieser Programme recht gut verkaufen. Und wer zum Beispiel das Schachprogramm oder den Flugsimulator für den ZX81 kennt, kann nur bestätigen, daß der »Kleinste« von Sinclair Qualitäten zeigen kann, die man ihm vorher nicht zugetraut hatte.

Mit dem Spectrum kam der Boom

Der große Durchbruch für Psion kam aber erst mit dem Spectrum. Einerseits hat Sinclair erkannt, daß die Software von Psion qualitativ sehr hoch stand, andererseits sah Psion die großen Marktchancen des Spectrums voraus (in England hat der Spectrum fast die gleiche Stellung wie der C64 hier in Deutschland). Dadurch kam es zu einer fruchtbaren Zusammenarbeit zwischen diesen beiden Firmen.

Neben solchen erfolgreichen Spielen wie Hungry Horace wurden auch wieder die Möglichkeiten des Computers mit Schach und Flugsimulator voll genutzt. Diese letzten beiden Programme werden wohl ebenso lange verkauft werden, wie der Spectrum erhältlich ist.

Durch die Strategie, eine Reihe ihrer Produkte unter dem Label Sinclair zu verkaufen, entstand für viele der Eindruck, Psion wäre mit Sinclair viel enger verbunden, als es tatsächlich der Fall ist. Psion geht sicher seine eigenen Wege. So hatte man sich bereits 1982 überlegt, die Firma nicht nur auf einem Bein, den Spielprogrammen, stehen zu lassen.

Psion begann bereits 1982 mit der Entwicklung eines Programm-Pakets für den professionellen Anwenderkreis. Es sollte die vier Hauptanwendungsbereiche, Textverarbeitung, Datenbank, Kalkulation und Grafik, umfassen und sich neben Leistungsfähigkeit durch äußerst einfache Bedienbarkeit auszeichnen.

Um ein solches Produkt in bestmöglicher Art herzustellen, benötigt man laut Potter die besten Hilfmittel, und so entschloß man sich bei Psion, zur Entwicklung der Programme eine »Mini Vax« anzuschaffen. Heute hat sich das Konzept, wie es scheint, derart bewährt, daß die Firma über vier dieser Computer verfügt.

Als Sinclair dann unter anderem auch an Psion herantrat, und nach den Möglichkeiten fragte, Anwendersoftware für den in der Planung befindlichen QL zu entwickeln, war Psion der Konkurrenz bereits einige Monate voraus. Alle QL-Besitzer kennen die Programme, an denen Psion in dieser Zeit gearbeitet hat. Es sind Quill, Abacus, Archive und Easel.

Hier erkennt man aber auch deutlich die Unabhängigkeit von Psion. Sinclair erhielt lediglich die Lizenz, diese Programme mit dem QL auf Microdrive zu vermarkten. Psion selber vermarktet das Produkt unter dem Namen XChange für andere Computer, zum Beispiel den IBM-PC und Kompatible, Apricot und andere. Auch gibt es in England einen weiteren Computer, der als Massenspeicher Microdrives (von Sinclair) benutzt, den OPD (One per Desk) von ITT, und auch hier wird XChange mit dem Gerät gleich mitgeliefert.

Interessant ist, daß XChange teurer ist, als der QL zusammen mit dem Software-Paket. Dabei kann man sagen, daß der Preis für XChange durchaus gerechtfertigt ist, betrachtet man Konkurrenzprodukte auf dem Markt für integrierte

Charme

Software. Dies unterstreicht Psions Ambitionen, die Nummer 1 unter den europäischen Softwarehäusern zu werden.

Neben dieser Betätigung auf dem Geschäftssektor gehört seit Mitte letzten Jahres auch die Herstellung und Vermarktung von Hardware zu Psions Aktivitäten. Es handelt sich um den Taschencomputer »Organizer«. Er ist programmierbar und kann bis zu 32 KByte Daten speichern. Diese Kapazität wird demnächst sogar auf 64 KByte ausbaubar sein.

Dabei handelt es sich um einen Permanentspeicher, die Daten bleiben sogar erhalten, wenn die Batterien ausgetauscht werden. Zusätzlich verfügt das Gerät über eine RS232-Schnittstelle und kann somit an alle Rechner, die eine solche Schnittstelle haben, angeschlossen werden.

Wen wundert es, daß sich auch dieses Gerät als Volltreffer erwies. In den ersten sechs Monaten wurden in England schon 20 000 Stück davon verkauft. Zur Zeit stellt Psion für den Organizer Datenbanken auf Modulen her, so gibt es bereits »Medipak«, eine Datenbank, die bei der medizinischen Krankheitsanalyse helfen kann.

Bei all diesen Aktivitäten auf dem Geschäftsbereich könnte man meinen, daß keine Zeit mehr für den Heimcomputermarkt übrig bleibt. Aber Psion hat kein Interesse, sich aus diesem zurückzuziehen. Man will lediglich die größeren Möglichkeiten der neuen Mikroprozessoren nutzen und sieht wohl auch eine große Zukunft in solchen modernen Computern wie dem QL (und anderen).

Daher ist es nicht verwunderlich, daß eines der ersten »Spielprogramme« für den QL auch aus dem Hause Psion stammt, das QL-Schach. Wiederum besticht das Programm durch Qualität. Dabei fasziniert nicht nur die hervorragende bewegte Grafik, sondern auch die Spielstärke des Programms. Zusammen mit reinen Schachcomputern konnte das QL-Schach im letzten Jahr Weltmeister der Schachcomputer-Weltmeisterschaften werden.

Ich frage Richard Lang, den Autor des Programms, ob er es jemals geschafft habe, gegen sein eigenes Werk zu gewinnen. Er lachte nur und meinte, er habe nicht die geringste Chance.

Es würde eigentlich zur Tradition von Psion gehören, einen Flugsimulator für den QL herzustellen. Jedoch gibt man nach außen hin keine Information über Programme, die nicht fertig sind. Auf meine Frage hin, wie ein möglicher Flugsimulator für den QL aussehen würde, gab es allerdings erstaunlich konkrete Ideen.

Psion ist eine Firma, die stark durch die Persönlichkeit ihres Gründers David Potter geprägt ist. Neben seiner wissenschaftlichen Qualifikation und dem Gespür für die Belange des Markts scheint er auch die Gabe zu haben, sich die geeigneten Mitarbeiter auszusuchen. Und er weiß nur zu genau, daß eine solche Firma nicht ohne diese Mitarbeiter den gewünschten Erfolg haben kann.

So ist es nicht verwunderlich, daß das Flaire, welches man spürt, wenn man die Räume dieser Firma betritt, außerordentlich wohltuend ist. David Frodsham, der die Abteilung Ausland leitet, führte mich durch alle Räume. Nirgendwo gab es einen Mitarbeiter, der nicht bereit war, seine Arbeit zu unterbrechen und zu zeigen, was gerade seine jetzigen Probleme beziehungsweise Projekte waren.

Irgendwie hat man den Eindruck, daß alle diese Leute wissen, warum sie bei Psion und nicht irgendwo anders arbeiten. Bemerkenswert war in diesem Zusammenhang David Potter's Äußerung, daß er eine Firma haben möchte, in der die Mitarbeiter ihre Erfüllung finden können.

David Frodsham drückte die Einstellung der Mitarbeiter wahrscheinlich sehr treffend aus, als er sagte »Wir arbeiten nicht gegen jemanden, wir arbeiten für Psion, denn Psion, daß sind wir!«



Der Hausherr von Psion: David Potter, ein Strahlemann der Branche

David Potter, der Mann hinter der Firma Psion

Computer hatten ihn seit je her interessiert. Ehe David Potter mit 36 Jahren die Firma Psion gründete, beschäftigte er sich mit Plasmaphysik und der Lösung der hier anfallenden Probleme mit Hilfe von Großrechnern. In entspannter Atmosphäre steht (oder besser sitzt) er bei einer Tasse Kaffee bereitwillig Rede und Antwort.

Frage: Wie ist es zur Gründung der Firma Psion gekommen?

David Potter: Schon immer habe ich viel darüber nachgedacht, was ich mit meinem Leben anfange, und ich würde es begrüßen, wenn man mehr als eines davon zur Verfügung hätte. Leider ist das nicht der Fall, und so muß man sich entschließen, wozu man sich entscheidet. Für mich gab es seit je her einmal die akademische Laufbahn und als Alternative den wirtschaftlichen Bereich. Durch mein Elternhaus lag es nahe,

an die Universität zu gehen, und für einige Jahre habe ich dort nach der Promotion Lehre und Forschung betrieben. Dabei hat mich die Welt der Computer fasziniert und ich habe einige Artikel unnd Bücher über den Einsatz von Computern in der Forschung geschrieben. Mit der Entwicklung der Mikrochips eröffnete sich für mich die Möglichkeit, auch mein zweites Interesse, eine eigene Firma zu haben, ohne auf den Umgang mit Computern verzichten zu müssen, zu verwirklichen.

Allerdings habe ich mir die Entscheidung, die akademische Laufbahn aufzugeben, nicht leicht gemacht. Ich muß gestehen, daß die Höhe der Gehälter für Forscher an englischen Universitäten auch einen Einfluß auf meine Entscheidung hatte.

Frage: Die Gründung einer Firma kostet doch auch einiges Geld. Hatten Sie diese Mittel zur Verfügung?

D. Potter: Ich hatte 1974, als ich anfing, konkret über eine Firma nachzudenken, 2000 Pfund auf meinem Konto. Dieses Geld habe ich in Aktien angelegt. Zu dieser Zeit war ich gerade an der Universität von Californien. Als ich nach England zurückkehrte, hatte sich mein Einsatz glücklicherweise verdreifacht. Das hat mich ermutigt, systematisch an der Börse Geld zu investieren. Als ich dann soweit war und 50000 Pfund zusammen hatte, nahm sich meine Frau das Geld und kaufte ein Haus.

Das war sicher gut so, da wir auch Kinder haben. Allerdings mußte ich mit der Börse wieder von vorne anfangen, um das Startkapital zu bekommen, die Firma zu gründen. Aus der Tatsache, daß es Psion gibt, können Sie erkennen, daß mir dies auch ein zweites Mal gelungen ist.

Frage: Wann waren Sie soweit?

D. Potter: Im Oktober 1980 wurde Psion gegründet.

Frage: Wieviele Mitarbeiter hatten Sie?

D. Potter: Angefangen hat Psion als Ein-Mann-Betrieb. Ich habe zunächst Produkte gesucht, die gut waren, aber wo es an der Vermarktung fehlte. Damit kam das erste Geld herein. So konnte ich daraufhin arbeiten, mein eigentliches Ziel zu erreichen, die Entwicklung von eigenen Produkten.

Nachdem im ersten Jahr 12000 Pfund Umsatz gebracht hatte und Charles Davies so kühn war, sich mir anzuschließen, konnten wir im nächsten Jahr schon einen Umsatz von 620000 Pfund erreichen. Sicher war die frühe Mitarbeit von Charles Davies sehr gut für die Firma. Ich kannte ihn sehr gut vom Imperial College hier in London, da er bei mir seine Doktorarbeit schrieb. Er war der beste Student, den ich kennengelernt habe.

Frage: Wie kam es zu der Zusammenarbeit mit Sinclair?

D. Potter: Anfangs waren wir für Sinclair ein Software-Haus wie jedes andere. Wir haben sicher sehr früh die guten Chancen des ZX81 erkannt und versucht, das Potential auch dieses kleinen Computers voll auszunutzen. Im Prinzip war es eine verrückte Idee, einen Flugsimulator für den ZX81 zu konzipieren. Aber das Projekt war sehr erfolgreich genauso wie das Schachprogramm. Sinclair hat dadurch umgekehrt erkannt, daß unsere Software dem Erfolg ihrer Produkte nur zugute kommt.

Frage: Wie steht es mit den Arcade-Games?

D. Potter: Ja, wir haben auch einige Arcade-Games produziert, aber das war nie unser Hauptanliegen. Vielmehr lag uns immer mehr an Spielen, bei denen der Benutzer auch ein wenig nachdenken muß. Nehmen Sie Scrabble, wir haben ein Programm entwickelt, welches einen Wortschatz von 12000 Worten im 48-KByte-Spectrum zur Verfügung stellt, und der durchschnittliche Spieler wird sicher vom Programm geschlagen.

Frage: Offensichtlich liegen Sie mit Ihrem Konzept nicht falsch.

D. Potter: Nein, ich glaube nicht. Schon im dritten Jahr des Bestehens von Psion konnten wir 1,5 Millionen Pfund umsetzen, und das bestätigt, daß unsere Produkte die Zustimmung der Benutzer haben. Wir sind jetzt bei einem Umsatz von 10 Millionen Pfund angelangt, wobei allerdings berücksichtigt werden muß, daß mittlerweile ein nicht unerheblicher Teil hiervon aus unseren Aktivitäten im professionellen Bereich herrühren.

Frage: Wie sieht die Zukunft von Psion aus?

D. Potter: Wir haben ein wichtiges Ziel erreicht, nämlich die Tatsache, daß wir die Firma auf solidem Grund fest etabliert haben. Dies geschah dadurch, daß wir uns neben dem reinen Spiele-Markt für Heimcomputer auch den professionellen Markt durch XChange und den Organizer geschlossen haben.

Durch die großen Umsatzzahlen sind wir in der Lage gewesen, verschiedene große Computer anzuschaffen, so daß wir hervorragende Hilfsmittel zur Entwicklung unserer Produkte zur Verfügung haben. Dies wird uns auf dem Weg helfen, die Nummer eins unter den europäischen Software Häusern zu werden.

Frage: Wie steht es mit dem amerikanischen Markt?

D. Potter: Natürlich haben wir auch diesen Markt im Auge. Es gibt bereits eine Abteilung Psion USA in den Staaten, und wir werden in naher Zukunft XChange dort vermarkten.

Frage: Mit all den Erfolgen auf dem Business-Sektor, ziehen Sie sich aus dem Heimcomputer-Markt völlig zurück?

D. Potter: Oh nein! Wir wissen sehr genau, womit wir groß geworden sind, und wir werden diesen Markt sicher nicht verlassen. Was wir in der Zukunft hier produzieren werden, können Sie am QL-Schach sehen.

In nicht allzu ferner Zukunft werden die sogenannten Heimcomputer noch wesentlich leistungsfähiger sein, wie man am QL ja sehen kann. Maschinen wie der Macintosh sind letztlich »Heimcomputer«. Für diese Computer gilt es, Software herzustellen, die den Möglichkeiten, die sie eröffnen, gerecht werden. Und Psion wird sich hier stark engagieren.

Frage: Welche Software wird Psion in 5 und in 10 Jahren produzieren?

D. Potter: Hoffentlich sehr viel! Wir glauben, daß auf dem Gebiet der künstlichen Intelligenz, was immer damit gemeint ist, sehr viel passieren wird. Nehmen Sie als Beispiel etwas, was ich gerne als Sprach-Prozessor bezeichne. Heute haben wir Textverarbeitungssysteme, und kaum jemand, der ein gutes System — wie Quill — benutzt hat, mag die Möglichkeit, die sich dadurch ergeben, noch missen.

Der nächste Schritt ist ein Programm, welches die Rechtschreibung überprüft. Nun, solche Programme gibt es bereits und sind eigentlich ein großes Problem, sobald nur genügend Speicherkapazität zur Verfügung steht. Dann kommt der Schritt, die Grammatik zu überprüfen. Auch das ist bereits möglich und auch Psion wird bald ein solches Programm auf den Markt bringen. Wenn man sich dies betrachtet, dann kann man das folgende auch bereits sehen, nämlich ein Programm, welches den Stil des Dokuments überprüft. Auch hier ist im Prinzip nur die Speicherkapazität eine notwendige Vorraussetzung.

Der Sprachprozessor ist dann verwirklicht, wenn der letzte Punkt verwirklicht ist, die Spracheingabe. Ich hoffe, daß Psion in sagen wir fünf Jah-

ren auf diesem Markt ein wichtiges Wort mit zu reden hat.

Aber das ist natürlich nicht das einzige. Die Computer in den neunziger Jahren werden immer »intelligenter«, neben der Spracherkennung wird es auch über eine Kamera die Bilderkennung geben und irgendwann werden die Computer uns sicher nicht in den Fähigkeiten nachstehen, die anfallenden Aufgaben zu erledigen.

Ob dies vom philosophischen oder theologischen Standpunkt aus gesehen sinnvoll ist, steht auf einem anderen Blatt. Wir wissen, daß der Mensch die Dinge verwirklicht, zu denen er fähig ist, und genauso wird es mit dem Bau von leistungsfähigeren Computern sein.

Frage: Eine letzte Frage: Sie verdienen mit Ihrer Firma sicher eine Menge Geld. Warum kaufen Sie sich nicht irgendwo in der Südsee ein Stück Land und setzen sich zur Ruhe?

D. Potter: Das ist langweilig! □

Die Multis kommen

Wer immer noch meint, Deutschland würde Ländern wie England und den USA in der Entwicklung von Hard- und Software für Homecomputer nachhinken, kann jetzt eines Besseren belehrt werden.

Inzwischen übersetzen deutsche Firmen nicht mehr gute englische Anwenderprogramme, sie schreiben sie selber — und das gar nicht so schlecht, wie das folgende Beispiel einer deutschen Firma zeigen wird, die inzwischen ein ganzes Paket leistungsstarker Soft- und Hardware für den Spectrum anbietet.

Angefangen hat alles mit Multifile, einem Adreßverwaltungsprogramm, das in Zusammenarbeit mit Tasword, Briefe selbständig adressiert und die Texte mit einer Anrede — wahlweise mit »Sehr geehrter Herr X, Frau Y oder Damen und Herren« — versieht.

Die Idee für dieses Programm entstand aus der eigenen Notlage Hunderte von Rundbriefen per Hand adressieren zu müssen. Da der englische Softwaremarkt in dieser Hinsicht nichts anbot, machte man aus der Not eine Tugend und entwickelte eine voll menügesteuerte und im Maschinencode geschriebene Adressendatei, die zudem noch sehr anwenderfrendlich ist, das durch die menügesteuerte Maske, das Lesen langweiliger und langwieriger Beschreibungen nahezu entfällt. Nach anfänglichen Schwierigkeiten, Datenverwaltung und richtige Anrede miteinander zu verknüpfen, ist nun auch dieses Problem gelöst. Das Programm sortiert selbständig nach Nachnamen; Anrede und Adressierung erfolgen trotzdem in richtiger Reihenfolge, zum Beispiel »Sehr geehrte Frau Mustermann« und nicht »Frau Mustermann, Erika«. Und wen die alte Version stört, kann sie, gegen einen geringen Aufpreis, gegen die neue Version umtauschen.

Fast die gleiche Maske wie bei Multifile findet sich auch bei Multilage. Hierbei handelt es sich um ein Artikel- und Lagerverwaltungssystem, das alle gebräuchlichen Informationen einer Warenverwaltung beinhaltet.

Erstellt wird sowohl eine Artikelliste mit Vekaufspreisen wie auch eine Inventurliste mit den dazugehörenden Einkaufspreisen. Außerdem werden Artikel, die den Mindestbestand unterschreiten, besonders ausgeworfen. Beide Listen können auch über den Drucker ausgegeben werden.

Auch bei Multilage ist die Bedienung sehr einfach, da menügesteuert, das Sort der pro Datei möglichen 532 Artikel sehr schnell.

Die dritte Datei in diesem Multi-Bund ist neu: Multidata — eine universelle Datei, bei der der Anwender seine Maske mit Hilfe eines Full-Screen-Editors selbst erstellen kann. Obligatorisch sind auch hier die 64 Zeichen/Zeile. Eine Bedienungsleiste im unteren Teil des Bildschirms zeigt an, mit welchen Tasten man Unterprogramme wie Suchen/Löschen anwählt. Auch hier ist das Suchen und Sortieren der Daten sehr schnell und erfolgt nach beliebigen Kriterien.

Die Daten von Multidata können durch ein Code-Wort verschlüsselt werden und sind für Unbefugte nicht einsehbar. Einen weiteren Vorteil bietet diese Datei durch die Möglichkeit, die Formatierung für den Drucker selbst zu bestimmen. Zur Verfügung stehen zwei Formate (Etiketten und Listen), die auch bei bereits erstellten Dateienform veränderbar sind, so daß mit gleichen Daten sowohl Etiketten wie auch Listenausdrucke möglich sind.

Alle drei Multi-Dateien werden auf Kassette geliefert, sind aber ohne Probleme direkt auf Diskette beziehungsweise Microdrive-Cartridge überspielbar und somit auf allen drei Datenträgern zu nutzen.

Doch Multi-Produkte beschränken sich nicht nur auf Dateien. Speziell für das Beta-Disc-Floppy-System wurde eine Reihe von Programmen neubearbeitet beziehungsweise neuentwickelt. Multitas ist zum Beispiel eine Erweiterung für das Tasword, bei der die Option für die Blockverschiebung stark verbessert und beschleunigt wurde. Auch das Suchen bestimmter Worte im Text ist jetzt wesentlich schneller.

Eine ähnliche Weiterentwicklung ist Multipas. Hier wurde das Hisoft Pascal an den Floppy-Betrieb angepaßt, wobei jetzt selbst die Include-Funktion auf Diskette möglich und DOS-Befehle aus dem Compiler aufrufbar sind. Ein besonderer Leckerbissen für alle Floppy-Besitzer dürfte der »Disketten-Doktor« Multidisc sein, daß an anderer Stelle in diesem Heft beschrieben wird.

Und daß demnächst deutsche Firmen auch auf dem Hardware-Sektor mitreden können, beweist das neueste Multi-Produkt: Multicom — eine Schnittstelle für Datenfernübertragung. Als absolute Weltneuheit und früher als englische Firmen es anbieten konnten, ist der Spectrum mit Hilfe einer RS232-Schnittstelle und der dazugehörenden Software sowie einem Akustik-Koppler jetzt in der Lage, mit anderen Computertypen fehlerfrei zu kommunizieren. Der Weg in alle Mailboxen des In- und Auslands ist frei, da sowohl die Übertragungsrate (7 oder 8 Bit) als auch Stop-Bits (1 oder 2) und Parity-Bits (gerade oder ungerade) frei einstellbar sind. Übertragen wird mit 300 Baud, allerdings sind als Option für den Export auch andere Übertragungsgeschwindigkeiten möglich.

Mit dieser Neuentwicklung ist also (fast) der letzte Schritt getan, das Image des Spectrum aufzupolieren. Außerdem zeigt das Beispiel auch: Es tut sich was, auch in diesem unseren Lande.

(Karina Krawczyk)

Markt&Technik-Buchverlag

An diesen Sinclair-Büchern kommen Sie nicht vorbei



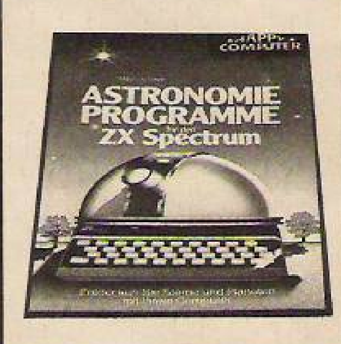
R. Egeler
ZX-Spectrum Hardware
Januar 1985, 147 Seiten
Dieses Buch vermittelt Ihnen ein fundiertes Basiswissen über Aufbau und Entwicklung eigener Hardware. Es gibt Ihnen die Möglichkeit, Ihren Spectrum professionell für Meß- und Regelungsaufgaben einzusetzen · Ausführliche Beschreibung der einzelnen ICs mit Abbildungen und 2-System-Schaltplänen · Anschluß einer PIO-Ansteuerung von Dezimalanzeigen · Leuchtdioden · Relais · DIL-Schalter · Eine akkugepufferte Hardwareuhr mit vierstelliger Anzeige · Soundgenerator mit drei Kanälen.
Best.-Nr. MT 737
(Sfr. 27,50/öS 232,40) **DM 29,80**



D. Laine
Maschinencode-Programme für den ZX Spectrum
1984, 204 Seiten
Nützliche Maschinencode-Programme mit Ihrem ZX Spectrum · Sortierung von Fließkommazahlen · Übernahme von Parametern direkt von einem Basic-Programm · Flußdiagramme · für Profis und solche, die es werden wollen.
Best.-Nr. MT 702
(Sfr. 29,50/öS 249,60) **DM 32,—**



T. Bridge/R. Carnell
ZX-Spectrum Abenteuerspiele
September 1984, 208 Seiten
Die Entstehungsgeschichte der Abenteuerspiele mit repräsentativen Beispielen für jede »Epoche«. Ein Programm speziell für Ihren ZX-Spectrum: »Das Auge des Sternenkriegers«, ein Grafik-Abenteuerspiel, das Sie in Atem hält!
Best.-Nr. MT 712
(Sfr. 27,50/öS 232,40) **DM 29,80**



M. Gavin
Astronomie-Programme für den ZX-Spectrum
September 1984, 268 Seiten
Eine phantastische Reise in die Welt des Kosmos mit Ihrem ZX-Spectrum: Der Julianische Kalender · Die Mondphasen · Eigene Satelliten starten · Kepler's Umlaufbahnen · Die Umlaufbahn Plutos · Interessant nicht nur für Hobby-Astronome.
Best.-Nr. MT 732
(Sfr. 27,50/öS 232,40) **DM 32,80**

Dr. E. Heß
Schnelles Rechnen mit dem ZX81
Oktober 1984, 276 Seiten
Das Betriebssystem · der BASIC-Interpreter · Gleitkomma-Macro-Befehle zur Verkürzung der Rechenzeiten · alle Programmbeispiele sind lauffähig auf dem ZX81 mit dem 1K-RAM-Speicher, ein 16K-Speicher vereinfacht die Programmentwicklung.
Best.-Nr. MT 706
(Sfr. 27,50/öS 232,40) **DM 29,80**

Einführung in die Mikrocomputertechnik. 1982, 4. überarb. Auflage, 488 Seiten
Das schon legendäre Standardwerk spiegelt den Stand der Mikrocomputertechnik wider · ausführlich erläuterte Grundbegriffe · alles über Speicherung von Informationen im Mikrocomputer · Vergleich alternativer Mikrocomputersysteme.
Best.-Nr. PW 138 **DM 66,— (Sfr. 60,70/öS 514,80)**

Basic für Einsteiger. 1984, 239 Seiten
Ein Arbeitsbuch für den absoluten Anfänger · Basic-Anweisungen Schritt für Schritt erklärt und anhand von einfachen Beispielen erläutert · das beliebte Arbeitsmittel für Lehrkräfte und für den interessierten Computerfan.
Best.-Nr. MT 680 **DM 32,— (Sfr. 29,50/öS 249,60)**

UCSD Pascal. September 1984, 492 Seiten
Eine Unterweisung in das weit verbreitete und auf allen gängigen Mikrocomputern verfügbare UCSD-Pascal · überschaubare Lerneinheiten · eingehende Behandlung der Datenstrukturen, Records und Dateien · numerische Verfahren für Statistiken · Sortieralgorithmen · die Maschinensprache.
Best.-Nr. MT 715 **DM 64,— (Sfr. 58,90/öS 499,20)**

Der Einstieg in FORTH. November 1984, 337 Seiten
Editieren von Programmen · Fehlersuche und -Korrektur · Diskettenoperationen · Zahlentypen · Grundlagen des strukturierten Programmierens · der FORTH-Standard FORTH-79 und Erweiterungen · ausführliches Glossar · FORTH — die Sprache für alle, die mehr aus ihrem Computer herausholen wollen!
Best.-Nr. MT 786 **DM 58,— (Sfr. 53,40/öS 452,40)**

Im Land der Abenteuer. 1984, 146 Seiten
Ein Lösungsbuch für zahlreiche Computerspiele · Tod in der Karibik · Transsylvanien · Unternehmen Asteroid · Das geheimnisvolle Haus · Zauberer und Prinzessin · Das goldene Vlies · Zeitzone · Der dunkle Kristall.
Best.-Nr. MT 699 **DM 29,80 (Sfr. 27,50/öS 232,40)**

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Fragen Sie dort nach unserem Gesamtkatalog mit über 170 neuen Computerbüchern.



Bestellkarten bitte an Ihren Buchhändler oder an einen unserer Depot-Händler. Adressenverzeichnis am Ende des Heftes. Beim Markt & Technik Verlag eingehende Bestellungen werden von den Depot-Händlern ausgeliefert.

Markt & Technik Verlag AG Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik-Vertriebs AG, Kollerstraße 3, CH-6300 Zug, ☎ 042/223155
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien ☎ 0222/677526

Die angegebenen Preise sind Ladenpreise.

Wir hoffen, daß diese Einführung Ihnen eine Hilfe ist. Ein komplettes Programm ausführlich weiter zu kommentieren, würde sicherlich den Rahmen einer üblichen Einführung sprengen.
Sollten Ihnen nach Kurs-Ende noch einige Fragen unbeantwortet geblieben sein, so schreiben Sie mir kurz. Ich werde dann versuchen, Ihnen weiterzuhelfen. Meine Anschrift finden Sie am Ende dieses Sonderheftes. Bitte fügen Sie Rückporto bei.

(Harald Wilhelm)

Routinen für die Praxis

Professionelle Programme für die Datenverarbeitung im Büro, in der Lagerverwaltung, für komplizierte Berechnungen sind überwiegend in Maschinensprache geschrieben. Nur dann laufen sie entsprechend schnell. Happy-Computer stellt, gewissermaßen aus dem Baukasten, einen Satz leistungsstarker Maschinencoderoutinen vor, mit denen ein Spectrum-Besitzer in der Lage sein sollte, auch Programme für ernsthafte Anwendungen zu schreiben.

Voraussetzung sind allerdings Erfahrungen in Maschinensprache und ein 48-KByte-Spectrum. Vom Prinzip her lassen sich die Routinen auf jeden anderen Z80-Computer, wie den Schneider CPC 464 oder den Epson QX10 übertragen. Der Computerbesitzer muß allerdings die Speicherbelegungen seines Computers kennen, und die Adressen ändern. Weiterhin müssen Sie einen Assembler besitzen, der Labels verarbeitet.

Die Palette der Routinen beginnt bei der Anlage von Stringarrays und führt über leistungsstarke Printroutinen zu einem variablen Sort. Der Abschluß wird, mit vielen Beispielen, der Kalkulator sein.

Auf einige assemblerspezifische Eigenschaften soll bereits zu Beginn hingewiesen werden:

DEFB	=	1. Byte wird übergeben
DEFW	=	2. Word wird übergeben
DEFM	=	Zeichen wird als Code übergeben
DEFS	=	Platzhalter
#	=	Hexzahl
EQU	=	Label definieren

Bevor wir uns an die erste Routine heranwagen, sollten einige kurze Voraussetzungen geschaffen werden. Der Befehl

CALL #OD6B

führt ein CLS aus. Anschließend muß der Bildschirm mit folgender Befehlsfolge eröffnet werden:

LD A,2
CALL #1601

Die Printposition auf dem Bildschirm wird in der Maschinensprache auf eine eigentümliche Art und Weise berechnet. Die Zeilen werden von oben (24) nach unten (1) gezählt. Die Spalten werden von links (33) nach rechts (2) gezählt.

Listing Nr	Art der Routine	Adresse > Start	— Ende
1	DIM-Befehl	EA60	EAA1
2	Deutscher Zeichensatz	EAA2	EE16
3	SCREEN$-Farben	EE17	EE31
4	PRINT-Routine	EE32	EE4B
5	Tastatur-Abfrage	EE4C	EEC3
6	String-Abfrage	EEC4	EEDB
7	Eingabefelder markieren	EEDC	EEE4
8	Feldmarkierungen löschen	EEE5	EEEA
9	Vergleich der Eingabe	EEEB	EEFD
10	Ausgabe von Datenfeldern	EEFE	EF06
11	Zahlenausgabe	EF07	EF6B
12	PLOT-/DRAW-Routine	EF6C	EF7A
13	PLOT-/DRAW-Routine	EF7B	EF89
14	PLOT-/DRAW-Routine	EF8A	EF98
15	PLOT-/DRAW-Routine	EF99	EFA7
16	SCROLL-Routine	EFA8	EFFC
17	Sortier-Routine	EFFD	F1D2
18	Zahlen-Umwandlung	F1D3	F241
19	String-Operation	F242	F27A
20	String-Operation	F27B	F29B
21	String-Operation	F29C	F2C6
22	Wandlung dezimal > hexadezimal	F2C7	F36C
23	LOAD-/SAVE-Routine	F36D	F37A
24	LOAD-/SAVE-Routine	F37B	F387
25	LOAD-/SAVE-Routine	F388	F3A8
26	LOAD-/SAVE-Routine	F3A9	F3C1

Liste der Maschinencode-Routinen

```
EA60                1        ORG   60000
                    2 ***
                    3 *** LISTING 1
                    4 ***
0028               10 ANZAHL EQU   40
0064               20 LAENGE EQU   100
5C4B               30 VARS   EQU   23627
EA60 2A4B5C        40 START  LD    HL,(VARS)
EA63 E5            50        PUSH  HL
EA64 01A80F        60        LD    BC,ANZAHL*LAENGE+8
EA67 CD5516        70        CALL  #1655
EA6A E1            80        POP   HL
EA6B E5            90        PUSH  HL
EA6C 36C1         100        LD    (HL),193
EA6E 23           110        INC   HL
EA6F 01A50F       120        LD    BC,ANZAHL*LAENGE+5
EA72 71           130        LD    (HL),C
EA73 23           140        INC   HL
EA74 70           150        LD    (HL),B
EA75 23           160        INC   HL
EA76 3602         170        LD    (HL),2
EA78 23           180        INC   HL
EA79 012800       190        LD    BC,ANZAHL
EA7C 71           200        LD    (HL),C
EA7D 23           210        INC   HL
EA7E 70           220        LD    (HL),B
EA7F 23           230        INC   HL
EA80 016400       240        LD    BC,LAENGE
EA83 71           250        LD    (HL),C
EA84 23           260        INC   HL
EA85 70           270        LD    (HL),B
EA86 23           280        INC   HL
EA87 3620         290        LD    (HL)," "
EA89 54           300        LD    D,H
EA8A 5D           310        LD    E,L
EA8B 13           320        INC   DE
EA8C 019F0F       330        LD    BC,ANZAHL*LAENGE-1
EA8F EDB0         340        LDIR
EA91 E1           350        POP   HL
EA92 11440F       360        LD    DE,ANZAHL-1*LAENGE+8
EA95 19           370        ADD   HL,DE
EA96 016500       380        LD    BC,LAENGE+1
EA99 3680         390 LOOP01 LD    (HL),128
EA9B 23           400        INC   HL
EA9C 0B           410        DEC   BC
EA9D 78           420        LD    A,B
EA9E B1           430        OR    C
EA9F 20F8         440        JR    NZ,LOOP01
EAA1 C9           450        RET
                  451 *L+
                  452 **
                  453 ** LISTING 2
                  454 **
5C36              460 CHARS  EQU   23606
EAA2 010003       470 SETCHA LD    BC,768
EAA5 21003D       480        LD    HL,15616
EAA8 11CFEA       490        LD    DE,CHARSE
EAAB EDB0         500        LDIR
EAAD 21CFED       510        LD    HL,NEWCHA
EAB0 5E           520 SETCH1 LD    E,(HL)
EAB1 23           530        INC   HL
EAB2 56           540        LD    D,(HL)
EAB3 23           550        INC   HL
EAB4 7A           560        LD    A,D
EAB5 B3           570        OR    E
EAB6 280A         580        JR    Z,SETCH2
```

Position beziehen

Um nun eine Position auf dem Bildschirm zu setzen, laden Sie die entsprechende Zeilennummer in das Prozeßregister B und die Spaltennummer in das Register C. Die Befehlsfolge

LD BC, #1821
CALL #0DD9

setzt nun die Printposition AT 0,0.

Ein Zeichen bringen wir mit folgender Folge auf den SCREEN:

LD A,"A"
RST 16

In diesem Fall wurde das Zeichen A in den Akkumulator gebracht, und mit dem zweiten Befehl gedruckt.

Da wir sehr oft Screenergebnisse erhalten werden, sollte man sich diese Befehle auf jeden Fall merken.

Dimensionierung

(Assemblerlisting 1)

Die erste Routine simuliert einen DIM-Befehl in Maschinensprache. Beschränkt wird die ganze Angelegenheit auf zweidimensionale Stringarrays.

Große Datenmassen sollen für den Programmierer schnell und einfach zugänglich sein. Ebenso ist die Voraussetzung vorgegeben, daß diese Datenfelder auch aus Basic anzusprechen sind. Was ist da einfacher, als die hauseigenen Spectrum-Variablen zu mißbrauchen.

String-Konstruktionen

Die vorgestellte Routine legt ein Stringarray an. Sie vergeben in Zeile 10 die Anzahl der Felder +1, und in Zeile 20 die Länge jedes Feldes. Das Addieren der 1 ist notwendig, da das ganze letzte File nicht benutzt werden kann. Das geschieht aus folgendem Grund:

Um das Ende der Datei zu erkennen, suchen wir im Speicherbereich das angehängte Stopbyte (128); siehe Spectrum-Manual Seite 165. Da dieses durch weitere, später angehangene Variablen, nach oben geschoben wird, würden wir das Ende der Datei nie finden. Aus diesem Grund belegen wir das letzte File mit 128. Irgendwo finden wir das Ende-Zeichen jetzt immer.

Die Funktionsweise ist eigentlich ganz einfach. Wir holen aus der Systemvariablen VARS die Anfangsadresse des Variablenbereiches.

An dieser Stelle beschaffen wir uns mit dem

CALL #1655

die Gesamtlänge, berechnet aus Anzahl*Länge+8, Platz. Die 8 Bytes sind Informationen an das Betriebssystem des Spectrum.

Wir schreiben nun in diese 8 Bytes folgende Daten:

Byte 1:	indizierter Variablenname. Hier wird der Code des Zeichens +128 eingeschrieben.
Byte 2: und 3:	Die folgenden 2 Bytes sind die Gesamtlänge, Berechnet aus Anzahl*Länge+5.
Byte 4:	Anzahl Dimensionen (2)
Byte 5: und 6:	Hier wird die Anzahl der Felder eingefügt.
Byte 7: und 8:	Hier wird die Länge eines Feldes eingetragen.

Mit diesen 8 Bytes erkennt das Betriebssystem sofort alle Syntaxfehler. Ebenfalls kann durch die Angabe der Gesamtlänge+5 dieser Variablenblock übersprungen werden. Unsere Routine füllt jetzt noch den gesamten restlichen Speicherbereich mit Blanks. Zu diesem Zweck wird der Schaufelbefehl LDIR benutzt. Zwischen Zeile 360 und 410 wird das letzte File mit 128ern gefüllt. In der letzten Zeile erfolgt der Rücksprung.

```
EAB8 0608       590         LD   B,8
EABA 7E         600 LOOP02  LD   A,(HL)
EABB 12         610         LD   (DE),A
EABC 23         620         INC  HL
EABD 13         630         INC  DE
EABE 10FA       640         DJNZ LOOP02
EAC0 18EE       650         JR   SETCH1
EAC2 21CFEA     660 SETCH2  LD   HL,CHARSE
EAC5 110001     670         LD   DE,256
EAC8 AF         680         XOR  A
EAC9 ED52       690         SBC  HL,DE
EACB 22365C     700         LD   (CHARS),HL
EACE C9         710         RET
EACF            720 CHARSE  DEFS 768
EDCF A7ED       730 NEWCHA  DEFW 91*8+CHARSE
EDD1 22001C02   740         DEFB 34,0,28,2,30,34,30,0
EDD9 AFED       750         DEFW 92*8+CHARSE
EDDB 22001C22   760         DEFB 34,0,28,34,34,34,28,0
EDE3 B7ED       770         DEFW 93*8+CHARSE
EDE5 22002222   780         DEFB 34,0,34,34,34,34,28,0
EDED BFED       790         DEFW 94*8+CHARSE
EDEF 00384478   800         DEFB 0,56,68,120,68,68,120,64
EDF7 A7EC       810         DEFW 59*8+CHARSE
EDF9 423C4242   820         DEFB 66,60,66,66,126,66,66,0
EE01 AFEC       830         DEFW 60*8+CHARSE
EE03 423C4242   840         DEFB 66,60,66,66,66,66,60,0
EE0B B7EC       850         DEFW 61*8+CHARSE
```

```
EE0D 42004242   860            DEFB 66,0,66,66,66,66,60,0
EE15 0000       870            DEFW 0
                871 *L+
                872 **
                873 ** LISTING 3
                874 **
0007            880 INK        EQU  7
0000            890 PAPER      EQU  0
0000            900 BORDER     EQU  0
EE17 CD4D0D     910 FARBE      CALL #0D4D
EE1A 3E10       920            LD   A,16
EE1C D7         930            RST  16
EE1D 3E07       940            LD   A,INK
EE1F D7         950            RST  16
EE20 CDAD1C     960            CALL #1CAD
EE23 3E11       970            LD   A,17
EE25 D7         980            RST  16
EE26 3E00       990            LD   A,PAPER
EE28 D7        1000            RST  16
EE29 CDAD1C    1010            CALL #1CAD
EE2C 3E00      1020            LD   A,BORDER
EE2E CD9B22    1030            CALL #229B
EE31 C9        1040            RET
               1041 *L+
               1042 **
               1043 ** LISTING 4
               1044 **
EE32 46        1050 PRINT1     LD   B,(HL)
EE33 23        1060            INC  HL
EE34 4E        1070            LD   C,(HL)
EE35 23        1080            INC  HL
EE36 E5        1090            PUSH HL
EE37 CDD90D    1100            CALL #0DD9
EE3A E1        1110            POP  HL
EE3B 46        1120            LD   B,(HL)
EE3C 23        1130            INC  HL
EE3D C5        1140 PRINT2     PUSH BC
EE3E 7E        1150            LD   A,(HL)
EE3F 23        1160            INC  HL
EE40 E5        1170            PUSH HL
EE41 D7        1180            RST  16
EE42 E1        1190            POP  HL
EE43 C1        1200            POP  BC
EE44 10F7      1210            DJNZ PRINT2
EE46 7E        1220            LD   A,(HL)
EE47 FE80      1230            CP   128
EE49 20E7      1240            JR   NZ,PRINT1
EE4B C9        1250            RET
               1251 *L+
               1252 ***
               1253 *** LISTING 5
               1254 ***
```

Aufgerufen wird diese Routine mit CALL START. Andere Variablen dürfen nur nach dem Aufruf definiert werden.

Nachdem wir nun ein Datenfeld anlegen können, wollen wir natürlich auch Menüs, Kommentare oder Fehlermeldungen auf den Bildschirm bringen. Diese sollen, da wir deutsch sprechen, auch die deutschen Umlaute enthalten.

Zeichensatz erneuern
(Assemblerlisting 2)

Diese Routine fertigt einen neuen Zeichensatz mit deutschen Umlauten an. Sie wurde so variabel geschrieben, daß Sie auch Zeichen umdefinieren können.

Wenn dieses Unterprogramm mit CALL SETCHA aufgerufen wird, kopiert es erst einmal den Originalzeichensatz in einen Bereich mit dem Namen CHARSE.

Sie liegt nun das erste Byte unserer Änderungen und prüft es auf Ende (0). Wenn hier keine 0 steht werden die nachfolgenden 8 Bytes in den mit der Adresse spezifizierten Bereich eingeschrieben. Anschließend wird die Systemvariable CHARS neu berechnet (CHARSE-256).

Wenn Sie mit eigenen Änderungen arbeiten, ziehen Sie vom ASCII-Code des zu ändernden Zeichens den Wert 32 ab. Schreiben Sie das Ergebnis als erste Zahl in das ansonsten identische DEFW Statement.

Ihre Vorschläge kommen in das DEFB Statement. Vergessen Sie nicht, am Ende die 0 anzuhängen.

Diese beiden Routinen sollten von Ihren Programmen immer zuerst aufgerufen werden. Der Bildschirm soll natürlich auch noch Farbe bekennen.

Die nachfolgende Kurzroutine nimmt Ihnen diese Arbeit ab.
(Assemblerlisting 3)

Diese Routine rufen Sie mit CALL FARBE auf. Bei Return entspricht die Bildschirmfarbe Ihren Wünschen.

Es gibt hier keinerlei Besonderheiten, die beachtet werden müssen. Diese Teile müßten eigentlich ohne große Erklärungen zu verstehen sein. Es werden lediglich die Spectrum-Steuerzeichen benutzt.
(Assemblerlisting 4)

Die nachfolgende Routine entstand, nachdem klar war, daß in kommerziellen Programmen ein ständiger Menü-Aufbau notwendig und somit unerläßlich ist und nachdem folgende Bedingungen an sie geknüpft wurden:

1. Die zu druckenden Zeichen sollen direkt auf eine, mit dem Ausdruck fest verbundene, Position gesetzt werden.
2. Das nötige CR muß angehängt werden können.
3. Die Zahl der Zeichen muß variabel sein.
4. Mehrere Ausdrücke sollen mit einem Aufruf gedruckt werden

Diese kurze Routine erfüllt diese Anforderungen. Allerdings soll das Datenfeld aus den oben genannten Gründen ein ganz bestimmtes Format haben:

```
DAT001 DEFW #1821     ; Position
       DEFB 6         ; Zeichenzahl
       DEFM "TEXT!"   ; Text
       DEFB 13        ; CR
       DEFB 128       ; Ende
```

Der Aufruf geschieht nun nach folgendem Muster:

```
LD HL,DAT001
CALL PRINT1
```

Sind die beiden ersten Bytes in das Doppelregister BC geladen, wird diese Position auf dem Bild-

```
1F3D                1260 PAUSE   EQU   #1F3D
EE4C 013200         1270 GET1    LD    BC,50
EE4F CD3D1F         1280         CALL  PAUSE
EE52 21B05C         1290         LD    HL,23728
EE55 3620           1300         LD    (HL)," "
EE57 FDCB01AE       1310         RES   5,(IY+1)
EE5B CD5FEE         1320         CALL  GET2
EE5E C9             1330         RET
EE5F CD71EE         1340 GET2    CALL  GET3
EE62 11B05C         1350         LD    DE,23728
EE65 32B05C         1360         LD    (23728),A
EE68 010000         1370         LD    BC,0
EE6B FE00           1380         CP    0
EE6D 28DD           1390         JR    Z,GET1
EE6F 0C             1400         INC   C
EE70 C9             1410         RET
EE71 3A075C         1420 GET3    LD    A,(#5C07)
EE74 FEFF           1430         CP    255
EE76 2003           1440         JR    NZ,GET4
EE78 3E00           1450         LD    A,0
EE7A C9             1460         RET
EE7B CD8E02         1470 GET4    CALL  #028E
EE7E 0E00           1480         LD    C,0
EE80 2802           1490         JR    Z,GET5
EE82 3E00           1500         LD    A,0
EE84 CD1E03         1510 GET5    CALL  #031E
EE87 3802           1520         JR    C,GET6
EE89 3E00           1530         LD    A,0
EE8B 15             1540 GET6    DEC   D
EE8C 5F             1550         LD    E,A
EE8D CD3303         1560         CALL  #0333
EE90 C9             1570         RET
EE91 CD4CEE         1580 GET     CALL  GET1
EE94 FEE2           1590         CP    226
EE96 2003           1600         JR    NZ,GET7
EE98 3E7B           1610         LD    A,123
EE9A C9             1620         RET
EE9B FEC3           1630 GET7    CP    195
EE9D 2003           1640         JR    NZ,GET8
EE9F 3E7C           1650         LD    A,124
EEA1 C9             1660         RET
EEA2 FECD           1670 GET8    CP    205
EEA4 2003           1680         JR    NZ,GET9
EEA6 3E7D           1690         LD    A,125
EEA8 C9             1700         RET
EEA9 FEC7           1710 GET9    CP    199
EEAB 2003           1720         JR    NZ,GET10
EEAD 3E5B           1730         LD    A,91
EEAF C9             1740         RET
EEB0 FEC9           1750 GET10   CP    201
EEB2 2003           1760         JR    NZ,GET11
EEB4 3E5C           1770         LD    A,92
EEB6 C9             1780         RET
EEB7 FEC8           1790 GET11   CP    200
EEB9 2003           1800         JR    NZ,GET12
EEBB 3E5D           1810         LD    A,93
EEBD C9             1820         RET
EEBE FECC           1830 GET12   CP    204
EEC0 C0             1840         RET   NZ
EEC1 3E7E           1850         LD    A,126
EEC3 C9             1860         RET
```

schirm gesetzt. Anschließend wird die Anzahl der Zeichen, einschließlich CR, in das Zählregister B geladen. Nun wird, bis B auf 0 ist, Zeichen für Zeichen geladen, und auf den Bildschirm gebracht.

Das dann folgende Byte ist entweder ein Stopbyte (128) oder die Zeilenposition des nächsten Ausdrucks.

Wenn Sie noch weitere Ausdrücke ein- oder anfügen wollen, so schieben Sie das Stopbyte immer an das Ende.

In meinen eigenen Programmen werden auf diese Art und Weise ganze Menüs auf den Bildschirm gebracht; und das verhältnismäßig schnell.

Wie der Computer mit Daten umgeht

Nun haben wir mit einem Ausdruck den Benutzer aufgefordert, eine Eingabe zu tätigen. Eine Tastaturabfrage muß her.

Die Tastaturabfrage

Die Tastaturabfrage (Listing 5) ist zugegebenermaßen das schwierigste Unterfangen. Diese Routine sollte selbständig auswerten und die Umlaute auf bestimmte Tasten legen. Die Routine führt die im Spectrum-ROM liegenden Auswertungsroutinen aus, und legt die Umlaute auf folgende Tasten:

ä = SYMBOL SHIFT a
ö = SYMBOL SHIFT s
ü = SYMBOL SHIFT d
ß = SYMBOL SHIFT f
Ä = SYMBOL SHIFT q
ö = SYMBOL SHIFT w
ü = SYMBOL SHIFT e

Wenn Sie sich ein wenig mit der Routine beschäftigt haben, können Sie die Umlaute auch auf Tasten legen, die Ihren Vorstellungen entsprechen.

Dieses Unterprogramm wird mit CALL GET aufgerufen. Sie springt nur bei einer gedrückten Taste zurück.

Der Code der gedrückten Taste befindet sich zur Weiterverarbeitung im Akkumulator.

Der Zeichenspeicher wurde in dieser Routine auf 23728 gelegt. Sollten Sie mit dem Beta-Disk-Floppylaufwerk oder mit dem Interface ZX-LPRINT III arbeiten, können bei älteren Versionen Komplikationen auftreten. Legen Sie in diesen Fällen den Zeichenspeicher auf einen anderen Platz.

Nun können wir mit unseren Programmteilen schon so allerhand. Was ist aber, wenn sie einen ganzen Satz als Eingabe erwarten? Schauen Sie in diesem Fall auf Routine 6.

String-Abfrage

Das im Listing 6 dargestellte Unterprogramm entstand aus folgender Überlegung:

```
                1861 *L+
                1862 ***
                1863 *** LISTING 6
                1864 ***
EEC4 C5         1870 INPUT1 PUSH BC
EEC5 E5         1880        PUSH HL
EEC6 CD91EE     1890        CALL GET
EEC9 FE0D       1900        CP   13
EECB 2809       1910        JR   Z,INPUT2
EECD E1         1920        POP  HL
EECE 77         1930        LD   (HL),A
EECF D7         1940        RST  16
EED0 C1         1950        POP  BC
EED1 23         1960        INC  HL
EED2 10F0       1970        DJNZ INPUT1
EED4 1802       1980        JR   INPUT3
EED6 C1         1990 INPUT2 POP  BC
EED7 C1         2000        POP  BC
EED8 3E0D       2010 INPUT3 LD   A,13
EEDA D7         2020        RST  16
EEDB C9         2030        RET
                2031 *L+
                2032 ***
                2033 *** LISTING 7
                2034 ***
EEDC 3E2E       2040 PUNKTE LD   A,"."
EEDE D7         2050        RST  16
EEDF 10FB       2060        DJNZ PUNKTE
EEE1 3E0D       2070        LD   A,13
EEE3 D7         2080        RST  16
EEE4 C9         2090        RET
                2091 *L+
                2092 ***
                2093 *** LISTING 8
                2094 ***
EEE5 3620       2100 TEILOE LD   (HL)," "
EEE7 23         2110        INC  HL
EEE8 10FB       2120        DJNZ TEILOE
EEEA C9         2130        RET
                2131 *L+
                2132 ***
                2133 *** LISTING 9
                2134 ***
EEEB 7E         2140 SUCTAB LD   A,(HL)
EEEC FE80       2150        CP   128
EEEE C8         2160        RET  Z
EEEF B9         2170        CP   C
EEF0 2007       2180        JR   NZ,SUCTA1
```

1. Eine bekannte Anzahl Zeichen soll von der Tastatur eingelesen werden.
2. Der Anwender soll nicht die maximale Zeichenanzahl eingeben müssen (CR als Abschluß).
3. Die eingegebenen Zeichen sollen direkt im Datenfeld gespeichert werden.
4. Die Zeichen sollen auf dem Bildschirm erscheinen.

Diese Routine erledigt das alles. Es handelt sich aber nicht um einen Bildschirmeditor. Daher soll immer eine Korrekturabfrage nachgeschoben werden. Aber dazu später.

Der CR soll nicht im Speicher abgelegt werden.

Der Einsprung in diese Routine ist aus den oben genannten Gründen natürlich an Bedingungen geknüpft. Aufgerufen wird sie also mit folgender Syntax:

```
LD HL,Adresse
LD B,Zeichenzahl
CALL INPUT1
```

Das Register HL enthält die Adresse des Speicherbereiches, in dem der Eintrag abgespeichert werden soll. Das Register B enthält die maximal erwartete Zeichenzahl.

Es ist sinnvoll die Adresse (HL) vor Aufruf der Routine zu retten. Dann addieren Sie bei Ende der Routine die maximale Größe des Eintrags zu der geretteten Adresse, und schon zeigt HL auf den neuen Speicherbereich.

Alle Register gezogen

Um nun Kommentare und Fehlermeldungen auf den Bildschirm zu bringen, suchen Sie sich einen Platz auf den Screen aus, auf dem solche Ausgaben erfolgen sollen.

Der eben erwähnte Korrekturaufruf könnte dann folgendes Format haben:

```
DAT002 DEFW Position
DEFB Länge, immer gleich
DEFM "Eintrag korrekt? (j/n)
DEFB 13
DEFB 128
LD     HL,DAT002
CALL   PRINT1
CALL   GET
CP     "n"
JR     Z,NICHT KORREKT
```

```
EEF2 23        2190        INC   HL
EEF3 5E        2200        LD    E,(HL)
EEF4 23        2210        INC   HL
EEF5 56        2220        LD    D,(HL)
EEF6 EB        2230        EX    DE,HL
EEF7 D1        2240        POP   DE
EEF8 E9        2250        JP    (HL)
EEF9 23        2260 SUCTA1 INC   HL
EEFA 23        2270        INC   HL
EEFB 23        2280        INC   HL
EEFC 18ED      2290        JR    SUCTAB
```

Immer wenn eine Eingabe erfolgte, lassen Sie eine solche Routine durchlaufen und verzweigen entweder nach "NICHT KORREKT" oder Sie gehen zur nächsten Eingabe über.

Aus diesem Grund empfahl ich eben, die Adresse auf dem Stack zu retten. Sobald die Eingabe falsch war, holen Sie die Adresse vom Stack, und starten diesen Eintrag neu.

Bei korrekter Eingabe addieren Sie eben nur die Eintragsbreite (wie oben bereits aufgeführt).

Für den Fall einer falschen Eingabe habe ich noch zwei kurze Routinen angefertigt.

Markieren von Eingabefeldern

Dieses Kurzprogramm (Listing 7) habe ich PUNKTE genannt. Da Sie in der Regel Eingabefelder auf dem Screen mit Punkten andeuten, müssen diese, vor allen Dingen bei einer Falscheingabe, wieder gelöscht werden können.
Der Aufruf erfolgt mit:

```
LD      B,Anzahl Punkte
CALL    PUNKTE
```

Als weitere Voraussetzung sollten Sie die Bildschirmposition der Punkte vorher bereits gesetzt haben.

Nachdem Sie mit CALL PUNKTE die Routine haben durchlaufen lassen, ist Ihr Eingabefeld wieder mit Punkten gelöscht.

Feldmarkierungen löschen

Es muß bei einer Falscheingabe natürlich auch der Speicher wieder korrigiert werden.

Das Unterprogramm TEILOE (Listing 8) ersetzt die durch B spezifizierte Anzahl Zeichen, an der Position in Register HL durch Blanks.

Da die beiden vorgenannten Routinen so einfach sind, erfolgt keine weitere Erläuterung.

Es kann natürlich auch sein, daß Sie dem Benutzer ein Menü einblenden, und eine Auswahl erwarten. In diesem Fall muß Ihr Maschinencode-Programm, genauso wie ein Basic-Programm, nach bestimmten Programmteilen verzweigen.

Vergleich der Eingabe

Dieser Routinevergleich der Eingabe (Listing 9) hat die Aufgabe, in einer Sprungtabelle nach einem Zeichen zu suchen, welches identisch mit einer Eingabe ist. Wenn dieses Zeichen gefunden wurde, soll eine Adresse geladen und direkt angesprungen werden.

Wenn das Zeichen nicht auffindbar ist, soll ein Return erfolgen, um eine Fehlermeldung drucken zu können.

Aus diesen Ausführungen haben Sie sicherlich geschlossen, daß einige Bedingungen mit dieser Routine verknüpft sind:

```
LD      HL,TAB001
LD      C,Suchzeichen
CALL    SUCTAB
*Rücksprung nur bei – Falsch!
```

Das Doppelregister HL enthält beim Einsprung die Basisadresse der Tabelle. Das eingegebene Zeichen, welches gesucht wird, soll sich in Register C befinden. Denken Sie an den Rücksprung. Verzweigen Sie hier wieder an Ihre Abfragestelle.

Die Tabelle muß ebenfalls ein ganz gestimmtes Format haben:

```
TAB001 DEFM "H"      *1. Suchzeichen
       DEFW Haupt    *Adresse dazu
       DEFM "E"      *1. Suchzeichen
       DEFW Eingab   *Adresse dazu
       DEFB 128      *Stopbyte
```

Zuerst kommt in der Tabelle ein zu vergleichendes Zeichen. Die dazu gehörige Adresse befindet sich dahinter. Nach der letzten Adresse muß das Stopbyte (128) stehen.

Am Anfang dieses Berichtes erwähnt ich, daß ein Assembler, der Labels verarbeitet, fast unverzichtbar ist. Sie haben Ihre Programmteile einfach mit "Haupt" oder "Eingab" benannt. Diese Labels setzen Sie in die Tabellen und schon geht alles wie von selbst.

Die Funktionsweise der Routine ist recht simpel.

Sie holt sich immer das erste Zeichen (Suchzeichen oder Ende), vergleicht es mit Ihrer Eingabe oder Ende. Bei Ende erfolgt ein Rücksprung. Bei falschem Zeichen wird der Zeiger HL um 3 erhöht, und die Routine neu gestartet.

Bei Übereinstimmung wird die Adresse in DE geladen. Diese Adresse wird duch Registertausch in HL gebracht, und die Rücksprungadresse vom Stapel geworfen. Als Abschluß wird die Routine direkt angesprungen.

Wenn wir aus unserem Datenfeld Einträge wieder herausholen wollen, so hilft uns die nächste Routine.

Ausgabe von Datenfeldern

Der Programmteil »Ausgabe von Datenfeldern« (Listing 10) wird mit CALL PRINTS aufgerufen.

Beim Einsprung muß das Register B die Anzahl der zu lesenden Zeichen, und das Register HL die Adresse des Eintrags enthalten. Ebenfalls muß die Bildschirmposition zu diesem Zeitpunkt bereits gesetzt sein.

```
LD HL,Adresse
LD B, Anzahl Zeichen
CALL PRINTS
```

Am Ende vollführt die Routine den notwendigen CR.

Es ist damit auch klar, warum sowohl der Spectrum, als auch meine Routinen ein Stringarray mit Blanks füllen. Sollte dies nicht der Fall sein, so würde jedes nicht vollgeschriebene Feld beim Druck eine Fehlermeldung bringen. Das CHR$ (0) wäre nicht zu interpretieren.

Bildschirm-Manipulation

Wir haben bisher die verschiedensten Dinge auf den Screen gebracht. Nun auch noch Zahlen.

Formatierte Zahlen

(Assemblerlisting 11)

Die nachfolgende Routine lehnt sich an ein Vorbild im Spectrum-ROM an. Dort existiert eine Routine (#1A28), die Zeilennummern in Dezimal wandelt und mit führenden Space, oder auch nicht, auf den Bildschirm bringt.

Meine Änderung bringt nun noch die folgenden Erweiterungen:

1. Wahlweiser Ausdruck von 3- oder 5stelligen Zahlen.
2. Es werden Nullen vorgestellt.

Damit können Sie ihre Zahlen formatieren. Es handelt sich hier um INTeger Zahlen. Floatingpointzahlen folgen im Kalkulatorabschnitt.

Der Einsprung folgt mit DE auf die Hexzahl zeigend. Bei 3stelligen Zahlen starten Sie mit CALL DEZIM2. 5stellige Zahlen werden mit CALL DEZIMA aufgerufen.

Diese Dezimalroutine zieht bei jedem Durchgang von der Zahl die jeweiligen Dezimalstufen ab. Wenn die Zahl kleiner ist wird eine "0" ausgegeben. Ansonsten wird die Zahl gedruckt.

Ich benutze diese Routine zum Ausdruck von Dateinummern, Mitgliedsnummern oder Artikelnummern.

Beim Einsprung muß die Bildschirmposition natürlich gesetzt sein.

Der Teil ab DEZIM1 entspricht fast vollständig dem ROM-Vorbild. Erweitert wurde dies allerdings um den Teil —10000. Der Spectrum braucht dies nicht, da Zeilennummern größer 9999 nicht verarbeitet werden.

Mit ein bißchen Phantasie läßt sich dieses Unterprogramm noch verkürzen. Das CR muß natürlich auch in diesem Fall angehangen werden.

Wir kommen nun zu 2 Bildschirmroutinen. Ursprünglich wollte ich solche nicht mit aufführen, denn in fast jedem Buch finden Sie diese. Aber meine sollten etwas besonderes sein.

PLOT und DRAW

(Assemblerlistings 12-15)

Diese vier Routinen sind als PLOT- oder DRAW-Routinen zu benutzen.

Die Pixeladressen werden im Spectrum, anders als die PRINT-Positionen, im Originalzustand in Maschinensprache benutzt.

Das heißt, gezählt wird von oben (175) nach unten (0), und von links (0) nach rechts (255).

Sie übergeben beim Aufruf in Register B die Anzahl der zu PLOTtenden Punkte. Das Doppelregister HL enthält die PLOT-Position.

Innerhalb von HL enthält L die X-, und H die Y-Koordinate.

Danach rufen sie die Routine auf, die Ihre gewünschte Richtung ausführt. Diese Routine ist nur geringfügig schneller als Basic.

Bei der nächsten Routine handelt es sich um eine umgeschriebene. In dem Buch »Maschinencode-Routinen für den ZX Spectrum« von Hardmann/Hewson, fand ich eine interessante SCROLL-Routine, die ich hier in einer abgeänderten Form vorstelle.

SCROLLen

(Assemblerlisting 16)

Der Effekt des SCROLLens sieht durch ein unter-den-Kopf scrollen natürlich besser aus.

Am Anfang wird überprüft, ob der Drucker angeschlossen ist. Wenn ja, wird zurückgesprungen. Wenn nein, wird das Scrollen ausgeführt.

Am Schluß der Routine wird auch die unterste Position wieder gesetzt, damit an dieser Stelle der nächste Eintrag erfolgen kann.

Die Funktionsweise ist eigentlich ganz simpel. Es wird jeweils das 8 Pixel tiefer liegende Byte in die aktuelle Adresse geladen. Dies geschieht 32 mal pro Zeile, 8 mal pro Zeichen, und 21 mal pro Zeile. Der Aufruf geschieht mit CALL SCROLL, ansonsten sind keine Bedingungen daran geknüpft.

Das Prüfen auf Druckbetrieb habe ich aus einem ganz bestimmten Grund mit eingebaut. Man kann Ausgaben auch in Maschinensprache so schreiben, daß sie für Drucker und Screen funktionieren. Bei Druckbetrieb funktioniert das Formatieren natürlich nicht. Deshalb wird dann zurückgesprungen. Das aufrufende Programm testet dann nochmals und verzweigt kurzfristig. Im Druckbetrieb muß die Anzahl der Ausgaben ja noch mitgezählt werden, da vor der Perforation ein FORM FEED erfolgen soll.

Basic kontra Code

So, und nun kommt ein ganz schönes Stück Arbeit: Ein Sort, aber ein besonderer. Es handelt sich um eine vom Basic in Maschinencode umgeschriebene Routine. Zuerst stelle ich Ihnen die Basic-Ursprungsroutine vor, und dann schauen Sie sich die zwar etwas langsame, aber leicht zu verstehende Maschinencode-Routine an. Selbst die Variablen habe ich größtenteils mit ihren Namen belassen.

Sehen Sie diese Routine nicht als "NON PLUS ULTRA", lassen Sie sich daran die Einfachheit von Maschinencode zeigen.

Sortieren in Basic

(Basiclisting 1)

Bevor Sie diese Routine anspringen können, müssen Sie in die Variable N die tatsächliche Anzahl der zu sortierenden Einträge eingeben. Ebenfalls sollten Sie mit X und Y die SLICE-Werte vergeben.

Wenn Sie A$() nur nach A$(i,10 TO 15) sortieren wollen, so übertragen Sie die 10 in X, und die 15 in Y.

Dieses Unterprogramm vergleicht nun den ersten Eintrag mit dem ersten Eintrag über der Hälfte. Ist der untere Eintrag kleiner oder gleich, so geschieht nichts. Sollte er aber größer sein, so werden die beiden ausgetauscht. Beide Zeiger werden um eins erhöht, und dann

```
8002 ***
8003 *** LISTING 1
8004 ***
8010 LET N=TATANZ
8020 LET M=N
8030 LET M=INT (M/2)
8040 IF M=0 THEN STOP
8050 LET J=1
8060 LET K=N-M
8070 LET I=J
8080 LET L=I+M
8090 IF A$(I,X TO Y)<=A$(L,X TO Y) THEN GOTO 160
8100 LET H$=A$(I)
8110 LET A$(I)=A$(L)
8120 LET A$(L)=H$
8130 LET I=I-M
8140 IF I<1 THEN GOTO 160
8150 GOTO 80
8160 LET J=J+1
8170 IF J>K THEN GOTO 30
8180 GOTO 70
```

Basic-Listing einer einfachen Sortier-Routine, die als Unterprogramm aufgerufen wird

```
                2361 *L+
                2362 ***
                2363 *** LISTING 11
                2364 ***
EF07 1A         2370 DEZIM2 LD    A,(DE)
EF08 6F         2380        LD    L,A
EF09 13         2390        INC   DE
EF0A 1A         2400        LD    A,(DE)
EF0B 67         2410        LD    H,A
EF0C 1821       2420        JR    DEZIM3
EF0E 1A         2430 DEZIMA LD    A,(DE)
EF0F 6F         2440        LD    L,A
EF10 13         2450        INC   DE
EF11 1A         2460        LD    A,(DE)
EF12 67         2470        LD    H,A
EF13 E5         2480        PUSH  HL
EF14 AF         2490        XOR   A
EF15 111027     2500        LD    DE,10000
EF18 ED52       2510        SBC   HL,DE
EF1A F24AEF     2520        JP    P,DEZIM1
EF1D 3E30       2530        LD    A,"0"
EF1F D7         2540        RST   16
EF20 E1         2550        POP   HL
EF21 E5         2560        PUSH  HL
EF22 AF         2570        XOR   A
EF23 11E803     2580        LD    DE,1000
EF26 ED52       2590        SBC   HL,DE
EF28 F24AEF     2600        JP    P,DEZIM1
EF2B 3E30       2610        LD    A,"0"
EF2D D7         2620        RST   16
EF2E E1         2630        POP   HL
EF2F E5         2640 DEZIM3 PUSH  HL
EF30 AF         2650        XOR   A
EF31 116400     2660        LD    DE,100
EF34 ED52       2670        SBC   HL,DE
EF36 F24AEF     2680        JP    P,DEZIM1
EF39 3E30       2690        LD    A,"0"
EF3B D7         2700        RST   16
EF3C E1         2710        POP   HL
EF3D E5         2720        PUSH  HL
EF3E AF         2730        XOR   A
EF3F 110A00     2740        LD    DE,10
EF42 ED52       2750        SBC   HL,DE
EF44 F24AEF     2760        JP    P,DEZIM1
EF47 3E30       2770        LD    A,"0"
EF49 D7         2780        RST   16
EF4A E1         2790 DEZIM1 POP   HL
EF4B 1EFF       2800        LD    E,255
EF4D 01F0D8     2810        LD    BC,#D8F0
EF50 CD2A19     2820        CALL  #192A
EF53 0118FC     2830        LD    BC,#FC18
EF56 CD2A19     2840        CALL  #192A
EF59 019CFF     2850        LD    BC,#FF9C
EF5C CD2A19     2860        CALL  #192A
EF5F 0EF6       2870        LD    C,#F6
EF61 CD2A19     2880        CALL  #192A
EF64 7D         2890        LD    A,L
EF65 CDEF15     2900        CALL  #15EF
EF68 3E0D       2910        LD    A,13
EF6A D7         2920        RST   16
EF6B C9         2930        RET
                2931 *L+
                2932 ***
                2933 *** LISTING 12
                2934 ***
EF6C C5         2940 PLOTRE PUSH  BC
EF6D 44         2950        LD    B,H
EF6E 4D         2960        LD    C,L
EF6F C5         2970        PUSH  BC
EF70 CDE522     2980        CALL  #22E5
EF73 C1         2990        POP   BC
EF74 0C         3000        INC   C
EF75 60         3010        LD    H,B
EF76 69         3020        LD    L,C
EF77 C1         3030        POP   BC
EF78 10F2       3040        DJNZ  PLOTRE
EF7A C9         3050        RET
                3052 ***
                3053 *** LISTING 13
                3054 ***
EF7B C5         3060 PLOTRU PUSH  BC
EF7C 44         3070        LD    B,H
EF7D 4D         3080        LD    C,L
EF7E C5         3090        PUSH  BC
EF7F CDE522     3100        CALL  #22E5
EF82 C1         3110        POP   BC
EF83 05         3120        DEC   B
EF84 60         3130        LD    H,B
EF85 69         3140        LD    L,C
EF86 C1         3150        POP   BC
EF87 10F2       3160        DJNZ  PLOTRU
EF89 C9         3170        RET
                3172 ***
                3173 *** LISTING 14
                3174 ***
EF8A C5         3180 PLOTLI PUSH  BC
EF8B 44         3190        LD    B,H
EF8C 4D         3200        LD    C,L
EF8D C5         3210        PUSH  BC
EF8E CDE522     3220        CALL  #22E5
EF91 C1         3230        POP   BC
EF92 0D         3240        DEC   C
EF93 60         3250        LD    H,B
EF94 69         3260        LD    L,C
EF95 C1         3270        POP   BC
EF96 10F2       3280        DJNZ  PLOTLI
EF98 C9         3290        RET
                3292 ***
                3293 *** LISTING 15
                3294 ***
EF99 C5         3300 PLOTRA PUSH  BC
EF9A 44         3310        LD    B,H
EF9B 4D         3320        LD    C,L
EF9C C5         3330        PUSH  BC
EF9D CDE522     3340        CALL  #22E5
EFA0 C1         3350        POP   BC
EFA1 04         3360        INC   B
EFA2 60         3370        LD    H,B
EFA3 69         3380        LD    L,C
EFA4 C1         3390        POP   BC
EFA5 10F1       3400        DJNZ  PLOTRA
EFA7 C9         3410        RET
                3411 *L+
                3412 ***
                3413 *** LISTING 16
                3414 ***
EFA8 FDCB014E   3420 SCROLL BIT   1,(IY+1)
EFAC C0         3430        RET   NZ
EFAD D9         3440        EXX
EFAE E5         3450        PUSH  HL
EFAF 1809       3460        JR    SCROL1
EFB1 E5         3470 DRITTE PUSH  HL
EFB2 210007     3480        LD    HL,1792
EFB5 19         3490        ADD   HL,DE
EFB6 EB         3500        EX    DE,HL
EFB7 E1         3510        POP   HL
EFB8 180A       3520        JR    ZEILE
EFBA 214040     3530 SCROL1 LD    HL,16448
EFBD 116040     3540        LD    DE,16480
EFC0 E5         3550 DRITT1 PUSH  HL
EFC1 D5         3560        PUSH  DE
EFC2 0E15       3570        LD    C,21
EFC4 0620       3580 ZEILE  LD    B,32
EFC6 1A         3590 UEBER  LD    A,(DE)
EFC7 77         3600        LD    (HL),A
EFC8 79         3610        LD    A,C
EFC9 E607       3620        AND   7
EFCB FE01       3630        CP    1
EFCD 2002       3640        JR    NZ,WEITER
EFCF 97         3650        SUB   A
EFD0 12         3660        LD    (DE),A
EFD1 23         3670 WEITER INC   HL
EFD2 13         3680        INC   DE
EFD3 10F1       3690        DJNZ  UEBER
```

```
EFD5 0D          3700             DEC  C
EFD6 2813        3710             JR   Z,NEU
EFD8 79          3720             LD   A,C
EFD9 E607        3730             AND  %00000111
EFDB FE00        3740             CP   0
EFDD 28D2        3750             JR   Z,DRITTE
EFDF FE07        3760             CP   7
EFE1 20E1        3770             JR   NZ,ZEILE
EFE3 D5          3780             PUSH DE
EFE4 110007      3790             LD   DE,1792
EFE7 19          3800             ADD  HL,DE
EFE8 D1          3810             POP  DE
EFE9 18D9        3820             JR   ZEILE
EFEB D1          3830 NEU         POP  DE
EFEC E1          3840             POP  HL
EFED 14          3850             INC  D
EFEE 24          3860             INC  H
EFEF 7C          3870             LD   A,H
EFF0 FE48        3880             CP   72
EFF2 20CC        3890             JR   NZ,DRITT1
EFF4 012103      3900             LD   BC,#0321
EFF7 CDD90D      3910             CALL #0DD9
EFFA E1          3920             POP  HL
EFFB D9          3930             EXX
EFFC C9          3940             RET
                 3941 *L+
                 3942 ***
                 3943 *** LISTING 17
                 3944 ***
0014             3950 TATANZ EQU  20
0028             3960 VON    EQU  40
003C             3970 BIS    EQU  60
EFFD 3E28        3980 SORT   LD   A,VON
EFFF 3221F0      3990        LD   (PAR_1),A
F002 3E3C        4000        LD   A,BIS
F004 3222F0      4010        LD   (PAR_2),A
F007 111400      4020 Z10    LD   DE,TATANZ
F00A ED5323F0    4030        LD   (VAR_N),DE
F00E 3E08        4040        LD   A,8
F010 3220F0      4050        LD   (ZAEHL),A
F013 2A4B5C      4060        LD   HL,(VARS)
F016 110800      4070        LD   DE,8
F019 19          4080        ADD  HL,DE
F01A 229CF0      4090        LD   (VAR_AN),HL
F01D C39EF0      4100        JP   SORT1
F020 00          4110 ZAEHL  DEFB 0
F021 00          4120 PAR_1  DEFB 0
F022 00          4130 PAR_2  DEFB 0
F023 0000        4140 VAR_N  DEFW 0
F025 0000        4150 VAR_M  DEFW 0
F027 0000        4160 VAR_J  DEFW 0
F029 0000        4170 VAR_K  DEFW 0
F02B 0000        4180 VAR_I  DEFW 0
F02D 0000        4190 VAR_L  DEFW 0
F02F             4200 VAR_H$ DEFS LAENGE
F093 00          4210 VAR_B  DEFB 0
F094 0000        4220 VAR_AI DEFW 0
F096 0000        4230 VAR_AL DEFW 0
F098 0000        4240 VAR_AJ DEFW 0
F09A 0000        4250 VAR_AM DEFW 0
F09C 0000        4260 VAR_AN DEFW 0
F09E 3A20F0      4270 SORT1  LD   A,(ZAEHL)
F0A1 FE00        4280        CP   0
F0A3 C8          4290        RET  Z
F0A4 3D          4300        DEC  A
F0A5 3220F0      4310        LD   (ZAEHL),A
F0A8 3A22F0      4320        LD   A,(PAR_2)
F0AB 2121F0      4330        LD   HL,PAR_1
F0AE 46          4340        LD   B,(HL)
F0AF 90          4350        SUB  B
F0B0 3C          4360        INC  A
F0B1 3293F0      4370        LD   (VAR_B),A
F0B4 ED5B23F0    4380 Z20    LD   DE,(VAR_N)
F0B8 ED5325F0    4390        LD   (VAR_M),DE
F0BC 1E00        4400 Z30    LD   E,0
F0BE 1602        4410        LD   D,2
F0C0 2A25F0      4420        LD   HL,(VAR_M)
F0C3 0608        4430        LD   B,8
F0C5 AF          4440 INTDIV XOR  A
F0C6 ED52        4450        SBC  HL,DE
F0C8 23          4460        INC  HL
F0C9 F2CEF0      4470        JP   P,INTDI1
F0CC 19          4480        ADD  HL,DE
F0CD 2B          4490        DEC  HL
F0CE 29          4500 INTDI1 ADD  HL,HL
F0CF 10F4        4510        DJNZ INTDIV
F0D1 CB4C        4520        BIT  1,H
F0D3 2801        4530        JR   Z,DIVFER
F0D5 2C          4540        INC  L
F0D6 2600        4550 DIVFER LD   H,0
F0D8 2225F0      4560        LD   (VAR_M),HL
F0DB ED5B25F0    4570 Z40    LD   DE,(VAR_M)
F0DF 7A          4580        LD   A,D
F0E0 B3          4590        OR   E
F0E1 CA9EF0      4600        JP   Z,SORT1
F0E4 110100      4610 Z50    LD   DE,1
F0E7 ED5327F0    4620        LD   (VAR_J),DE
F0EB AF          4630 Z60    XOR  A
F0EC 2A23F0      4640        LD   HL,(VAR_N)
F0EF ED5B25F0    4650        LD   DE,(VAR_M)
F0F3 ED52        4660        SBC  HL,DE
F0F5 2229F0      4670        LD   (VAR_K),HL
F0F8 ED5B27F0    4680 Z70    LD   DE,(VAR_J)
F0FC ED532BF0    4690        LD   (VAR_I),DE
F100 2A9CF0      4700 Z80A   LD   HL,(VAR_AN)
F103 ED4B2BF0    4710        LD   BC,(VAR_I)
F107 0B          4720        DEC  BC
F108 78          4730        LD   A,B
F109 B1          4740        OR   C
F10A 2005        4750        JR   NZ,Z80B
F10C 2294F0      4760        LD   (VAR_AI),HL
F10F 1811        4770        JR   Z80D
F111 ED4B2BF0    4780 Z80B   LD   BC,(VAR_I)
F115 116400      4790        LD   DE,LAENGE
F118 0B          4800        DEC  BC
F119 0B          4810 Z80C   DEC  BC
F11A 19          4820        ADD  HL,DE
F11B 78          4830        LD   A,B
F11C B1          4840        OR   C
F11D 20FA        4850        JR   NZ,Z80C
F11F 2294F0      4860        LD   (VAR_AI),HL
F122 3A21F0      4870 Z80D   LD   A,(PAR_1)
F125 3D          4880        DEC  A
F126 1600        4890        LD   D,0
F128 5F          4900        LD   E,A
F129 19          4910        ADD  HL,DE
F12A 2298F0      4920        LD   (VAR_AJ),HL
F12D 2A2BF0      4930 Z80    LD   HL,(VAR_I)
F130 ED5B25F0    4940        LD   DE,(VAR_M)
F134 19          4950        ADD  HL,DE
F135 222DF0      4960        LD   (VAR_L),HL
F138 2A9CF0      4970        LD   HL,(VAR_AN)
F13B ED4B2DF0    4980        LD   BC,(VAR_L)
F13F 0B          4990        DEC  BC
F140 78          5000        LD   A,B
F141 B1          5010        OR   C
F142 2005        5020        JR   NZ,Z80a
F144 2296F0      5030        LD   (VAR_AL),HL
F147 1811        5040        JR   Z80b
F149 ED4B2DF0    5050 Z80a   LD   BC,(VAR_L)
F14D 116400      5060        LD   DE,LAENGE
F150 0B          5070        DEC  BC
F151 0B          5080 Z80c   DEC  BC
F152 19          5090        ADD  HL,DE
F153 78          5100        LD   A,B
F154 B1          5110        OR   C
F155 20FA        5120        JR   NZ,Z80c
F157 2296F0      5130        LD   (VAR_AL),HL
F15A 3A21F0      5140 Z80b   LD   A,(PAR_1)
F15D 3D          5150        DEC  A
F15E 1600        5160        LD   D,0
F160 5F          5170        LD   E,A
F161 19          5180        ADD  HL,DE
F162 229AF0      5190        LD   (VAR_AM),HL
```

```
F165 2193F0     5200 Z90    LD   HL,VAR_B
F168 46         5210        LD   B,(HL)
F169 2A9AF0     5220        LD   HL,(VAR_AM)
F16C ED5B98F0   5230        LD   DE,(VAR_AJ)
F170 1A         5240 Z90A   LD   A,(DE)
F171 BE         5250        CP   (HL)
F172 2805       5260        JR   Z,Z90D
F174 300A       5270        JR   NC,Z100
F176 DABAF1     5280        JP   C,Z160
F179 23         5290 Z90D   INC  HL
F17A 13         5300        INC  DE
F17B 10F3       5310        DJNZ Z90A
F17D C3BAF1     5320        JP   Z160
F180 112FF0     5330 Z100   LD   DE,VAR_H$
F183 2A94F0     5340        LD   HL,(VAR_AI)
F186 016400     5350        LD   BC,LAENGE
F189 EDB0       5360        LDIR
F18B ED5B94F0   5370 Z110   LD   DE,(VAR_AI)
F18F 2A96F0     5380        LD   HL,(VAR_AL)
F192 016400     5390        LD   BC,LAENGE
F195 EDB0       5400        LDIR
F197 ED5B96F0   5410 Z120   LD   DE,(VAR_AL)
F19B 212FF0     5420        LD   HL,VAR_H$
F19E 016400     5430        LD   BC,LAENGE
F1A1 EDB0       5440        LDIR
F1A3 2A2BF0     5450 Z130   LD   HL,(VAR_I)
F1A6 ED5B25F0   5460        LD   DE,(VAR_M)
F1AA AF         5470        XOR  A
F1AB ED52       5480        SBC  HL,DE
F1AD 222BF0     5490        LD   (VAR_I),HL
F1B0 2A2BF0     5500 Z140   LD   HL,(VAR_I)
F1B3 CB7C       5510        BIT  7,H
F1B5 2803       5520        JR   Z,Z160
F1B7 C32DF1     5530 Z150   JP   Z80
F1BA 2A27F0     5540 Z160   LD   HL,(VAR_J)
F1BD 23         5550        INC  HL
F1BE 2227F0     5560        LD   (VAR_J),HL
F1C1 AF         5570 Z170   XOR  A
F1C2 2A27F0     5580        LD   HL,(VAR_J)
F1C5 ED5B29F0   5590        LD   DE,(VAR_K)
F1C9 ED52       5600        SBC  HL,DE
F1CB 2803       5610        JR   Z,Z180
F1CD F2BCF0     5620        JP   P,Z30
F1D0 C3F8F0     5630 Z180   JP   Z70
```

werden die beiden zweiten verglichen.

Das Verfahren läuft durch, bis der untere Zeiger die Hälfte erreicht. In diesem Fall wird die Hälfte wiederum halbiert. Wenn dabei der Wert 0 herauskommt, ist die Routine zu Ende.

Ich habe die Zeilennummern mit einem vorgestellten B aufgeschrieben, damit keiner versucht, dieses Listing in den Assembler einzutippen. Im Basic muß natürlich auch das B entfernt werden.

Was macht nun unser Maschinencode-Sort?

Maschinencode-SORT

(Assemblerlisting 17)

Genau dasselbe!

Vor Einsprung mit CALL SORT, vergeben Sie in TATANZ ebenfalls die tatsächliche Anzahl zu sortierender Einträge. Die beiden SLICER-Werte benennen Sie mit VON und BIS. Da ich davon ausgehe, daß Sie den Maschinencode-Sort im gleichen Programm benutzen wie die DIM-Routine (Nr. 1), brauchen Sie die Länge eines Eintrags nicht mehr zu vergeben. Diesen Wert entnehme ich aus dieser Routine.

Die tatsächliche Anzahl ist hier wichtiger als im Basic-Sort. Denn über das tiefer sortieren der Blanks hinaus, kann uns hier passieren, daß unser letztes File mit den Stopbytes (128) untergebuttert wird. In der Regel sollte das ja nicht geschehen, denn wer hat als Telefonnummer oder ähnlich einen TOKEN; aber man kann ja nie wissen!

Die tiefer sortierten Blanks würden bei Mitberücksichtigung den Such- und Änderbetrieb unserer Programme ganz schön verlängern. Aus diesem Grund lassen wir das.

Ich möchte an dieser Stelle nochmals daraufhin weisen, daß es beileibe schnellere und kürzere Sorts auf dem Markt gibt. Als Beispiel sei hier das "Beta Basic" genannt.

Bei schnelleren Programmen werden unter anderem die Indexregister als Zeiger benutzt. Es kann auch der Kalkulator benutzt werden.

Um die Übertragung von Basic in Maschinencode zeigen zu können, habe ich die Zeilennummern von Z10 bis Z180 mit in den Maschinencode-Sort genommen.

Arbeit im Untergrund

Der Kalkulator ist ein mächtiges Betriebssystem im Betriebssystem. Es gibt dem Anwender die ganze Gewalt des Rechnens und Kalkulierens. Darüber hinaus sind auch sämtliche Stringoperationen mit ihm ausführbar. Der Kalkulator wird mit dem Befehl

RST 40

aufgerufen. Diesem Befehl folgen einzelne Bytes, die jeweils eine ganze Operation einleiten. Der Kalkulator beginnt bei #335B. Die Obcodes der einzelnen Bytes beginnen bei #32D7. Ebenfalls können 5 verschiedene Konstanten (#32B5) jeweils mit einem Byte auf den Kalkulatorstack gebracht werden.

Eine Operation wird mit den auf dem Kalkulator-Stack befindlichen Ausdrücken durchgeführt. Ein Ausdruck besteht aus 5 Bytes. Diese liegen jeweils zwischen den Systemvariablen STKBOT und STKEND. Diese 5 Bytes können Stringparameter enthalten

Byte 1:	**0 für einen String von einem Array oder Teilstring**
	1 für einen einzelnen String
Byte 2:	**Lowbyte der Startadresse des Strings**
Byte 3:	**Highbyte**
Byte 4:	**Lowbyte der Länge des Strings**
Byte 5:	**Highbyte**

oder auch eine INTegerzahl:

Byte 1:	**0**
Byte 2:	**0 für positiv**
	255 für negativ
Byte 3:	**Lowbyte der INTegerzahl**
Byte 4:	**Highbyte**
Byte 5:	**0**

Ebenso können Floatingpointzahlen enthalten sein:

Byte 1:	**Vorzeichen und Exponent**
Byte 2:	**Mantissenbyte**
Byte 3:	**Mantissenbyte**
Byte 4:	**Mantissenbyte**
Byte 5:	**Mantissenbyte**

Sie sagen nun sicherlich, in dieser Form gebe ich doch keine Zahlen ein!

Das ist richtig: Wenn Sie eine Basic-Zeile in das Listing übergeben, führt der Spectrum einen Teil-RUN für diese Zeile durch. Ein Ergebnis kommt dabei zwar nicht zustande, aber die Syntax wird überprüft. Wenn nun in der Zeile eine Zahl dargestellt wird, besorgt sich das Betriebssystem 6 Bytes Platz; genau hinter Ihrer Zahl, die ja nur im ASCII-Code da steht. In das 1. Byte des Platzes setzt der Spectrum den Zahlenmerker (14) hin. Die 5 Bytes dahinter werden von einer Routine (#2C9B) umgewandelt. Entweder in das INTeger- oder Floatingpointformat.

```
                    5631 *L+
                    5632 ***
                    5633 *** K1
                    5634 *** LISTING 18
                    5635 ***
5C5D                5640 CHADD   EQU   23645
5C65                5650 STKEND  EQU   23653
F1D3 D5             5660 WANDEL  PUSH  DE
F1D4 225D5C         5670         LD    (CHADD),HL
F1D7 7E             5680         LD    A,(HL)
F1D8 CD9B2C         5690         CALL  #2C9B
F1DB 2A655C         5700         LD    HL,(STKEND)
F1DE 010500         5710         LD    BC,5
F1E1 AF             5720         XOR   A
F1E2 ED42           5730         SBC   HL,BC
F1E4 22655C         5740         LD    (STKEND),HL
F1E7 D1             5750         POP   DE
F1E8 EDB0           5760         LDIR
F1EA C9             5770         RET
                    5772 ***
                    5773 *** B1
                    5774 ***
000F                5780 ADD     EQU   #0F
0003                5790 SUB     EQU   #03
0004                5800 MUL     EQU   #04
0005                5810 DIV     EQU   #05
00A4                5820 KONS10  EQU   #A4
F1EB 312E3233       5830 ZAHL1   DEFM  "1.2345"
F1F1 0D             5840         DEFB  13
F1F2 322E3334       5850 ZAHL2   DEFM  "2.3456"
F1F8 0D             5860         DEFB  13
F1F9                5870 FREI1   DEFS  5
F1FE                5880 FREI2   DEFS  5
F203 CD6B0D         5890 EINSPR  CALL  #0D6B
F206 3E02           5900         LD    A,2
F208 CD0116         5910         CALL  #1601
F20B 012118         5920         LD    BC,#1821
F20E CDD90D         5930         CALL  #0DD9
F211 21EBF1         5940         LD    HL,ZAHL1
F214 11F9F1         5950         LD    DE,FREI1
F217 CDD3F1         5960         CALL  WANDEL
F21A 21F2F1         5970         LD    HL,ZAHL2
F21D 11FEF1         5980         LD    DE,FREI2
F220 CDD3F1         5990         CALL  WANDEL
F223 21F9F1         6000         LD    HL,FREI1
F226 ED5B655C       6010         LD    DE,(STKEND)
F22A 010500         6020         LD    BC,5
F22D EDB0           6030         LDIR
F22F 21FEF1         6040         LD    HL,FREI2
F232 010500         6050         LD    BC,5
F235 EDB0           6060         LDIR
F237 ED53655C       6070         LD    (STKEND),DE
```

Wenn jetzt ein Listing ausgegeben wird, bringt der Spectrum Zeichen für Zeichen auf den Bildschirm. Trifft er dabei auf die 14, so überspringt er diese und die nachfolgenden 5 Bytes.
Genial nicht?

Die umgewandelten Bytes benutzt er aber jetzt nur noch für alle Operationen.

Eine Routine, die unsere Zahlen umwandelt, folgt nun.

Kalkulator-Routine 1
(Assemblerlisting 18)

Das jetzt folgende Unterprogramm wurde extra so geschrieben, daß Sie von den Spectrum-Variablen unabhängig sind, und statt dessen Ihre eigenen Zahlen benutzen können. Sie müssen nur folgendes beachten:

1. **Halten Sie 2 Speicherbereiche mit jeweils 5 Bytes frei.**
2. **Schreiben Sie hinter die umzuwandelnden Zeilen grundsätzlich ein CR (13).**
3. **Der Einsprung erfolgt mit Register DE auf einen freien Platz zeigend, und**
4. **mit Register HL auf die umzuwandelnde Zahl zeigend.**

Diese Routine übersetzt Ihre Zahl und legt das Ergebnis in den freien Bereich. Den Kalkulator-Stack setzt sie wieder auf normal.

Da eine Operation nur mit 2 Einträgen abläuft, brauchen wir auch nur 2 Zwischenspeicher. Bei unseren künftigen Ausrechnungen belassen wir das jeweilige Zwischenergebnis auf dem Stack, und schieben nur den nächsten Operator nach.

Schauen Sie sich im Zusammenhang mit der Routine K1 auch das Beispiel B1 an. Das Byte ADD (#0F) ist einer dieser Opcodes. Die beiden Zahlen auf dem Stack werden addiert. Das Byte #38 signalisiert dem Kalkulator das Ende. Wenn Sie nun das Byte ADD durch #03 ersetzen werden die beiden Zahlen subtrahiert. Bei #04 wird multipliziert und bei #05 dividiert. Lassen Sie das Beispiel mit #05 mal laufen und schieben zwischen diesem und dem Ende noch ein #A4 und #04 ein. Ihr Ergebnis wird jetzt noch mit der Konstanten 10 multipliziert.
Überzeugt?

Der CALL #2DE3 bringt übrigens eine Floatingpointzahl auf dem Stack auf den Bildschirm.

Die Anzahl der Opcodes ist größer als die mit dem Interface 1 dabei gekommenen Hookcodes. Aus diesem Grund füge ich eine Übersicht aller Codes in der Anlage bei. Sie können sich diese aus dem Heft reißen oder besser kopieren und beim Arbeiten mit dem Kalkulator neben sich legen.

Bis jetzt haben wir nur mit Zahlen gearbeitet. Nun kommen Strings.

Kalkulator-Routine 2
(Assemblerlisting 19)

Statt mit im Speicher stehenden Variablen, arbeiten wir jetzt mit Strings. Wir stellen uns vor, irgendwo in unseren Datenfeldern stehen zu vergleichende Strings.

Nachdem was wir im ersten Kalkulator-Kapitel gehört haben, müßten wir auf den Stack deren Daten legen. Diese Arbeit nimmt uns eine Routine bei #2AB6 ab. Sie legt die Daten nach folgendem Format auf den Kalkulator-Stack:

Register A:	0
Register BC:	Länge des Strings
Register DE:	Startadresse des Strings

Nachdem wir mit dieser Routine die Daten der beiden Strings auf den Stack gelegt haben, springen wir, ohne Kalkulatoraufruf, eine seiner Routinen direkt an. Diese Routine (#353B) führt alle Stringoperationen durch. Beim Einsprung muß das Register B den Opcode enthalten.

Wenn das Ergebnis wahr ist, legt die Routine eine 1 als letztes Ergebnis auf den Stack. Bei unwahr wird eine 0 abgelegt. Mit unserer alten FP-Druckroutine an #2DE3 holen wir das Ergebnis auf den Screen.

Unser Programm führt nun folgenden Vergleich durch:
STR_1 < STR_2 Ergebnis = 1

Wenn Sie in STR_1 das A durch ein C ersetzen würden, käme eine 0 als Ergebnis. Versuchen Sie es ruhig. Der Aufruf erfolgt mit CALL STRVER.

Diese Routine ist schon mehr ein Beispiel, da die ganzen Bildschirmeröffnungsriten mit untergebracht sind.

Haben Sie es ausprobiert?

Der Vergleich würde der gleiche sein, aber das Ergebnis ist diesmal anders.

Die für diese Kalkulatorroutinen zulässigen Opcodes heißen:

#09 — #OE
#11 — #16

Wenn Sie mit anderen Opcodes einspringen erzielen Sie kein »gesundes« Ergebnis. Genauso verhält es sich auch mit den anderen Opcodes beim Einsprung in den Kalkulator.

Kalkulator-Routine 3
(Assemblerlisting 20)

Das bißchen Arbeit, das wir eben in der Routine noch investiert haben, kann uns natürlich wieder eine ROM-Routine abnehmen. Diese liegt an Adresse #24FB und kann unseren Ausdruck selbständig be-

```
F23B EF        6080         RST   #28
F23C OF        6090         DEFB  ADD
               6191 *L+
               6192 ***
               6193 *** K2
               6194 *** LISTING 19
               6195 ***
F242 41424344  6200 STR_1   DEFM  "ABCDEF"
F248 42434445  6210 STR_2   DEFM  "BCDEFG"
F24E CD6BOD    6220 STRVER  CALL  #0D6B
F251 3E02      6230         LD    A,2
F253 CD0116    6240         CALL  #1601
F256 012118    6250         LD    BC,#1821
F259 CDD90D    6260         CALL  #0DD9
F25C 3E00      6270         LD    A,0
F25E 1142F2    6280         LD    DE,STR_1
F261 010600    6290         LD    BC,6
F264 CDB62A    6300         CALL  #2AB6
F267 3E00      6310         LD    A,0
F269 1148F2    6320         LD    DE,STR_2
F26C 010600    6330         LD    BC,6
F26F CDB62A    6340         CALL  #2AB6
F272 0615      6350         LD    B,#15
F274 CD3B35    6360         CALL  #353B
F277 CDE32D    6370         CALL  #2DE3
F27A C9        6380         RET
               6381 *L+
               6382 ***
               6383 *** K3
               6384 *** LISTING 20
               6385 ***
F27B 41243E42  6390 OPE_1   DEFM  "A$>B$"
F280 OD        6400         DEFB  13
F281 CD6BOD    6410 OPERAT  CALL  #0D6B
F284 3E02      6420         LD    A,2
F286 CD0116    6430         CALL  #1601
F289 012118    6440         LD    BC,#1821
F28C CDD90D    6450         CALL  #0DD9
F28F 217BF2    6460         LD    HL,OPE_1
F292 225D5C    6470         LD    (CHADD),HL
F295 CDFB24    6480         CALL  #24FB
F298 CDE32D    6490         CALL  #2DE3
F29B C9        6500         RET
               6501 *L+
               6502 ***
               6503 *** K4
               6504 *** LISTING 21
               6505 ***
F29C 41243D41  6510 STRI1   DEFM  "A$=A$+B$"
F2A4 OD        6520         DEFB  13
F2A5 CD6BOD    6530 KOM     CALL  #0D6B
F2A8 3E02      6540         LD    A,2
```

```
F2AA CD0116     6550          CALL #1601
F2AD 012118     6560          LD   BC,#1821
F2B0 CDD90D     6570          CALL #0DD9
F2B3 219CF2     6580          LD   HL,STRI1
F2B6 225D5C     6590          LD   (CHADD),HL
F2B9 CD1F1C     6600          CALL #1C1F
F2BC E7         6610          RST  #20
F2BD CD561C     6620          CALL #1C56
F2C0 011400     6630          LD   BC,20
F2C3 CD6A1E     6640          CALL #1E6A
F2C6 C9         6650          RET
F23D 38         6100          DEFB #38
F23E CDE32D     6110          CALL #2DE3
F241 C9         6120          RET
                6130 ***
                6140 *** ERGAENZUNG ZU B1
                6150 ***
                6160 *** 6090 DEFB DIV
                6170 *** 6092 DEFB KONS10
                6180 *** 6094 DEFB MUL
                6190 ***
```

rechnen. Beim Rücksprung liegt wieder eine 0 für falsch auf dem Stack, oder eine 1 für richtig.

Bevor wir nun noch komplexe Operationen durchführen, sollten wir uns einmal über die Anwendung dieser Kalkulationsmöglichkeiten in unseren Programmen Gedanken machen.

Wir wollen eine Datei anlegen, in der zum Beispiel unsere finanziellen Mittel aufgeführt werden. Wir legen ein DIM-Feld (Routine 1) nach folgendem Muster an:
Text: 10 Zeichen
Betrag: 10 Zeichen = 999999.99 + CR

Unser DIM macht diesen Platz frei, aber die Aufgabe, an den letzten Platz ein CR zu setzen, bleibt uns überlassen. Das allein reicht aber nicht. Dieses Feld muß mit Nullen ausgefüllt werden; denn Zahlen mit Blank gibt es nicht.

Wir holen mit INPUT unsere Eingaben rein, und stellen sie vor unser Carriage Return.

Sobald wir die Reihe addieren wollen, holen wir jede Zahl auf den

Byte	Adresse	Auswirkung Dez. se Hex
0	368F	Sprung wenn der letzte Eintrag < > 0 ist. In diesem Fall enthält das nächste Byte einen relativen Sprungwert von 0 — 255. Die entsprechenden Bytes werden bei wahr übersprungen. Bei unwahr wird nur das nächste Byte geschluckt.
1	343C	Die letzten beiden Werte tauschen.
2	33A1	Der letzte Wert wird gelöscht.
3	300F	Subtraktionsroutine (auch FP).
4	30CA	Multiplikationsroutine (auch FP).
5	31AF	Divisionsroutine (auch FP).
6	3851	Hoch 2 (^2).
7	351B	ODER. Es liegen 2 Parameter auf dem Stack. Bei X OR Y ergibt das X, wenn Y=0; sonst ist X=1.
8	3524	Logisch UND. X AND Y ergibt X, wenn Y< >0; sonst ist X=0.
9	353B	ungleich.
10	353B	kleiner als.
11	353B	Strings sind ungleich.
12	353B	nicht größer.
13	353B	nicht kleiner.
14	353B	Strings sind ungleich.
15	3014	Additionsroutine (auch FP).
16	352D	String AND Nummer. A$ AND X ergibt A$, wenn X< >0; sonst wird A$ ein Nullstring.
17	353B	String < =.
18	353B	String > =.
19	353B	Strings < >.
20	353B	String >.
21	353B	String <.
22	353B	String =.
23	359C	Addition von Strings. Die neuen Parameter sind auf Stack.
24	35DE	VAL$
25	34BC	USR$
26	3645	Der aktuelle Kanal wird eröffnet, ein Zeichen geholt, ein Platz besorgt und die Parameter dieses Eintrags auf dem Stack gespeichert.
27	346E	Der letzte Eintrag wird negiert.
28	3669	CODE. Stringparameter vorher auf Stack legen.
29	35DE	VAL
30	3674	LEN. Stringparameter vorher auf Stack.
31	37B5	SINus
32	37AA	COSinus
33	37DA	TANgens
34	3833	ASN (Arcussinus)
35	3843	ACS (Arcuscosinus)
36	37E2	ATN (Arcustangens)
37	3713	LN (logarithmus)
38	36C4	EXP (E hoch X)
39	36AF	INTeger
40	384A	SQR (Wurzel)
41	3492	SGN (Sign = Vorzeichen)
42	346A	ABSolut
43	34AC	PEEK. INTegerzahl ist auf Stack.
44	34A5	IN. INTegerzahl ist auf Stack.
45	34B3	USR (Aufruf)
46	361F	STR$
47	35C9	CHR$
48	3501	NOT
49	33C0	Verdoppeln
50	36A0	N MOD M-Division
51	3686	Sprung
52	33C6	Parameter auf Stack
53	367A	Nachgebauter DJNZ Befehl
54	3506	< 0
55	34F9	> 0
56	369B	Ende Calculator
57	3783	Hole Agrument
58	3214	Ploatingpointzahl kappen und runden
59	33A2	Der Calculator im Calculator (FP-Calc-2)
60	2D4F	INTeger in Floatingpointzahl wandeln
61	3297	zurück auf Stack
62	3449	Entwicklung der Polynome
63	341B	0 auf Stack
64	342D	Register in MEM Bereich speichern
65	340F	MEM zurück in Register
		Die Konstanten
160	341B	0 auf Stack legen
161	341B	1 auf Stack legen
162	341B	0.5 auf Stack legen
163	341B	PI/2 auf Stack legen
164	341B	10 auf Stack legen
		Speichern von Stackinhalten in MEMO — MEM5
192	342D	1. Zahl oder Parametersatz = 5 Bytes
193	342D	2. dto.
194	342D	3. dto.
195	342D	4. dto.
196	342D	5. dto.
197	342D	6. dto.
		Laden von Parametern oder Zahlen aus MEMO — MEM5 in Stack
224	340F	1. Zahl oder Parametersatz = 5 Bytes
225	340F	2. dto.
226	340F	3. dto.
227	340F	4. dto.
228	340F	5. dto.
229	340F	6. dto.

Tabelle zum Abschnitt »Kalkulator«

Stack und addieren. Der Reihe nach:

1. erste Zahl holen
2. umwandeln
3. auf Stack
4. zweite Zahl holen
5. umwandeln
6. auf Stack
7. addieren
8. nächste Zahl holen
9. umwandeln
10. auf Stack
11. Sprung nach 7.

Zwischen jeder Operation prüfen wir den Anfang unseres Files (1. Zeichen vom Text) auf Blank, wenn ja wird dieser Eintrag übersprungen.

Geprüft muß natürlich auch das Ende der Datei werden. Wenn ja, Ende mit Ausgabe des Gesamtergebnisses.

Bei veränderlichen Zahlen, sollten Sie nur mit dem 5-Byte-FP-Format arbeiten.

Die letzten Feststellungen klingen zugegebenermaßen etwas kompliziert, werden aber an einem späteren Beispiel erläutert.

Die letzte Routine aus dem Kalkulatorkasten soll uns in die Lage versetzen, ganze Operationen mit einem Befehl ausführen zu lassen.

Kalkulator-Routine 4

(Assemblerlisting 21)

Diese Routine, die Sie mit CALL KOM starten, führt eine Operation eines Basic-Textes durch. Bevor Sie diese laufen lassen, sollten Sie schon die Variablen A$ und B$ definieren.

Nach den üblichen Bildschirmeröffnungen wird die Systemvariable CHADD auf den Beginn der Operation gesetzt. Der CALL #1C1F springt direkt in die Variablenbestimmung im ROM. Wir wissen zu diesem Zeitpunkt, daß es sich um eine LET-Zeile handelt, daher springen wir direkt ein.

Anschließend wird mit RST #20 das nächste Zeichen eingeladen (in diesem Fall das »=«) und mit diesem in die Zuweisungsroutine an #1C56 gesprungen.

Sie sollten im Basic vor dem Start eine Zeile 20 kreieren,

20 STOP

da wir in diese Zeile mit dem Call #1E66A hüpfen. Der Grund hierfür liegt in dem veränderten CHADD-Zeiger. Würde die Routine mit RET abgeschlossen, so hätten Sie eine Fehlermeldung vor Augen.

An der Adresse #1E6A finden Sie übrigens die GOTO Ausführung.

An dieser Stelle wollen wir einmal mit dem Kalkulator stoppen. Sie sollten die in diesem Kapitel aufgeführten Routinen als kleine, eigenständige Programme sehen (mit Ausnahme Routine K1, welche Ihre Zahlen aufbereitet).

Diese Beispiele können sehr aufschlußreich sein, und Ihnen viele Anregungen geben. Experimentieren Sie mit den Kalkulator Opcodes, und stellen Sie die verschiedenen Ergebnisse gegenüber. Die Routinen K2 bis K4, und das Beispiel B1 brauchen nicht in das Assembler-File, welches in Ihrer Programme implementiert werden kann, aufgenommen zu werden.

Dafür gehören noch die folgenden Routinen dort hinein.

Zahlenspiele, Laden und Speichern

(Assemblerlisting 22)

Erinnern Sie sich noch an die Routine, welche die HEX-DEZ-Wandlung vornahm, und das Ergebnis in Dezimal ausgab?

Ein Gegenstück stellt das folgende Unterprogramm dar. Es holt eine 3- oder 5stellige Dezimalzahl von der Tastatur und legt das Ergebnis in HEX ab. Einen großen Unterschied gibt es dennoch. Bei der Dezimalroutine sind nur 3- oder 5stellige Zahlen erlaubt. Bei der Eingaberoutine sind bis zu 5stellige Zahleneingaben erlaubt.

Der Einsprung für die 5stellige Zahl erfolgt mit CALL MULTIP, mit CALL ZU100 kann eine 3stellige Zahl übergeben werden. Das Ergebnis dieser Eingabe liegt bei RETurn in dem Speicherbereich MULT2. Bevor Sie allerdings in diese Routine einspringen, müssen Sie, wie fast überall, die Bildschirmposition bestimmen, denn die Ausgabe wird natürlich auch gesteuert.

An dieser Stelle sei nochmals bemerkt, daß es sich hierbei auch nicht um einen Bildschirmeditor handelt. Die Fehlerabfrage muß auch hier von Ihnen übernommen werden. Eine fehlerhafte Eingabe wird halt neu gestartet.

Nachdem Sie das Ergebnis für Ihre Bearbeitung aus MULT2 entnommen haben, sollten Sie diese 2 Bytes auf 0 stellen, da sonst bei einem weiteren Aufruf immer dazuaddiert würde.

Die Arbeitsweise wurde hier ebenfalls recht simpel gehalten. Der jeweilige Faktor wird in den Zwischenspeicher MULT geladen. Nachdem dies erledigt ist, wird eine Zahl von der Tastatur geholt. Bei CR wird zurückgesprungen. Ansonsten wandeln wir mit AND 15 die ASCII-Zahl in die Dezimalzahl und prüfen auf 0. Bei einer vorgestellten 0 springen wir zum nächsten Faktor.

Bei einer Zahl wird diese beim Unterprogramm MULTI mit dem Faktor multipliziert und gespeichert. Dann kommt der nächste Faktor.

Die Multiplikationroutine ist nichts anderes als eine Addition mit der Häufigkeit Ihrer eingegebenen Ziffer.

Diese Routine läßt sich auch kürzer schreiben, ist dann aber nicht mehr so einfach zu erläutern.

Der Faktorspeicher MULT und der Ziffernspeicher MULT1 werden am Ende gelöscht. Der Ergebnisspeicher MULT2 muß, wie bereits erwähnt, von Ihnen gelöscht werden.

LOAD und SAVE

(Assemblerlistings 23-26)

Mit diesen Unterprogrammen steht Ihnen ein Satz Lade- und Speicherroutinen zur Verfügung. Aufgerufen werden diese Teile mit dem jeweiligen Namen:

LOAD	:	Lade kopflos
SAVE	:	Speicher kopflos
LOAD_H	:	Lade mit Kopf
SAVE_H	:	Speicher mit Kopf

Bevor Sie eine der Routinen anspringen, sollten Sie erst folgende Werte vergeben:

PLACE:	Startadresse des zu ladenden oder speichernden Bereiches
LENGTH:	Anzahl der Bytes
HEADER:	Startadresse des 17-Byte-Headers

Der Bereich des Headers wird wie folgt belegt:

01:	Typ 0 = Basic 1 = Zahlenarray 2 = Buchstabenarray 3 = Code
02—11:	Name
12/13:	Länge
14/15:	Startadresse
16/17:	Programmlänge

Diese Belegung muß von Ihnen vorgenommen werden. Bei Laden oder Speichern mit Kopf, führen die beiden entsprechenden Routinen eine »PAUSE 200« zwischen HEADER und Daten aus.

Die ROM-Routinen benötigen eine Spezifikation vom HEADER. Wenn ein HEADER bearbeitet wird, muß das Register A eine 0 enthalten. Sollte es sich um einen Datensatz handeln, so steht im Akkumulator der Wert 255.

Wir wollen nun diese Routinen in der Praxis erproben. Ich versuche Ihnen einige Hinweise dazu zu ge-

```
EA60                1         ORG   60000
                 6651 *L+
                 6652 ***
                 6653 *** LISTING 22
                 6654 ***
F2C7 0000        6660 MULT    DEFW  0
F2C9 00          6670 MULT1   DEFB  0
F2CA 0000        6680 MULT2   DEFW  0
F2CC 211027      6690 MULTIP  LD    HL,10000
F2CF 22C7F2      6700         LD    (MULT),HL
F2D2 CD91EE      6710         CALL  GET
F2D5 FE0D        6720         CP    13
F2D7 C8          6730         RET   Z
F2D8 CD65F3      6740         CALL  MULTIQ
F2DB 2809        6750         JR    Z,ZU1000
F2DD 32C9F2      6760         LD    (MULT1),A
F2E0 CD56F3      6770         CALL  MULTI
F2E3 22CAF2      6780         LD    (MULT2),HL
F2E6 21E803      6790 ZU1000  LD    HL,1000
F2E9 22C7F2      6800         LD    (MULT),HL
F2EC CD91EE      6810         CALL  GET
F2EF FE0D        6820         CP    13
F2F1 C8          6830         RET   Z
F2F2 CD65F3      6840         CALL  MULTIQ
F2F5 2809        6850         JR    Z,ZU100
F2F7 32C9F2      6860         LD    (MULT1),A
F2FA CD56F3      6870         CALL  MULTI
F2FD 22CAF2      6880         LD    (MULT2),HL
F300 216400      6890 ZU100   LD    HL,100
F303 22C7F2      6900         LD    (MULT),HL
F306 CD91EE      6910         CALL  GET
F309 FE0D        6920         CP    13
F30B C8          6930         RET   Z
F30C CD65F3      6940         CALL  MULTIQ
F30F 2809        6950         JR    Z,ZU10
F311 32C9F2      6960         LD    (MULT1),A
F314 CD56F3      6970         CALL  MULTI
F317 22CAF2      6980         LD    (MULT2),HL
F31A 210A00      6990 ZU10    LD    HL,10
F31D 22C7F2      7000         LD    (MULT),HL
F320 CD91EE      7010         CALL  GET
F323 FE0D        7020         CP    13
F325 C8          7030         RET   Z
F326 CD65F3      7040         CALL  MULTIQ
F329 1809        7050         JR    ZU1
F32B 32C9F2      7060         LD    (MULT1),A
F32E CD56F3      7070         CALL  MULTI
F331 22CAF2      7080         LD    (MULT2),HL
F334 CD91EE      7090 ZU1     CALL  GET
F337 FE0D        7100         CP    13
F339 C8          7110         RET   Z
```

ben, ohne ein komplettes Programm für Sie zu erstellen. Wir lassen einige Routinen nacheinader ablaufen, und Sie erhalten von mir die Idee einer Adressenverwaltung.

Die Praxis

Wie Sie vielleicht bemerkt haben, sind die Routineneinsprünge verschieden gestaltet. Bei einigen müssen Register die notwendigen Informationen mitbringen, während bei anderen Routinen mit Variablen, definiert mit EQU, gearbeitet wird. Sie können dies abändern, sollten aber die Informationen, egal auf welche Art, auf jeden Fall übergeben.

Speichern Sie unsere Routinen als ein komplettes Textfile mit dem Assembler ab. Bevor Sie dies erledigen, löschen Sie aber die Zeile 1 mit dem ORG-Statement um dieses File später mit der Include-Möglichkeit in Ihr Programm einbauen zu können.

Wir wollen folgende Eingaben vorsehen:

Name	:	15 Zeichen
Strasse	:	15 Zeichen
Wohnort	:	15 Zeichen
Telefon	:	15 Zeichen

Zu diesem Zweck legen wir ein DIM-Feld A$(100,60) an. Ersetzen Sie im Listing 1 die 40 in Zeile 10 durch den Wert 101, und in Zeile 20 den Eintrag 100 durch 60. Mit einem CALL START haben wir nun unseren Speicherbereich geschaffen:

```
A001 ORG 55000
A002 *** RAMTOP SETZEN
A003 BEGINN LD HL,54999
A004 LD (23730),HL
A005 *** KLEINSCHREIBUNG
A006 LD HL,23658
A007 RES 3,(HL)
A008 CALL START
```

Ein Anfang wäre hiermit gemacht. Nun meinen Sie aber nicht, daß das Listing so ausführlich weiterführt. Ich helfe Ihnen nur über die Anfangsschwierigkeiten.

Als nächstes schaffen wir unseren deutschen Zeichensatz

```
A009 CALL SETCHA
A010 CALL FARBE
A011 CALL #*D6B
A012 LD A,2
A013 CALL #1601
```

und die Bildschirmfarben. Jetzt fängt es an schwierig zu werden. Es geht an das Menü. Wir rufen es folgendermaßen auf:

```
A014 HAUPT LD HL,DAT001
A015 CALL PINT1
```

Das zugehörige Datenfeld legen wir in ein separates Includefile ab. Wenn wir unser Routinenfile »R« genannt haben, so können wir dieses DATAfeld »B« nennen. Das Programmfile nennen wir »A«.

```
F33A F5        7120        PUSH AF
F33B D7        7130        RST  16
F33C F1        7140        POP  AF
F33D E60F      7150        AND  %00001111
F33F 2ACAF2    7160        LD   HL,(MULT2)
F342 5F        7170        LD   E,A
F343 1600      7180        LD   D,0
F345 19        7190        ADD  HL,DE
F346 3E00      7200        LD   A,0
F348 32C9F2    7210        LD   (MULT1),A
F34B 110000    7220        LD   DE,0
F34E ED53C7F2  7230        LD   (MULT),DE
F352 22CAF2    7240        LD   (MULT2),HL
F355 C9        7250        RET
F356 3AC9F2    7260 MULTI  LD   A,(MULT1)
F359 47        7270        LD   B,A
F35A 2ACAF2    7280        LD   HL,(MULT2)
F35D ED5BC7F2  7290        LD   DE,(MULT)
F361 19        7300 MULT3  ADD  HL,DE
F362 10FD      7310        DJNZ MULT3
F364 C9        7320        RET
F365 F5        7330 MULTIQ PUSH AF
F366 D7        7340        RST  16
F367 F1        7350        POP  AF
F368 E60F      7360        AND  %00001111
F36A FE00      7370        CP   0
F36C C9        7380        RET
               7381 *L+
               7382 ***
               7383 *** LISTING 23
               7384 ***
751C           7390 HEADER EQU  29980
7530           7400 PLACE  EQU  30000
0064           7410 LENGTH EQU  100
F36D 37        7420 LOAD   SCF
F36E 3EFF      7430        LD   A,255
F370 DD213075  7440        LD   IX,PLACE
F374 116400    7450        LD   DE,LENGTH
F377 CD5605    7460        CALL #0556
F37A C9        7470        RET
               7472 ***
               7473 *** LISTING 24
               7474 ***
F37B 3EFF      7480 SAVE   LD   A,255
F37D DD213075  7490        LD   IX,PLACE
F381 116400    7500        LD   DE,LENGTH
F384 CDC204    7510        CALL #04C2
F387 C9        7520        RET
               7522 ***
               7523 *** LISTING 25
               7524 ***
F388 37        7530 LOAD_H SCF
```

```
B001    DAT001 DEFW #1821
B002    DEFB 16
B003    DEFM "1 = Adresse neu"
B004    DEFB 13
B005    DEFW #1721
B006    DEFB 11
B007    DEFM "2 = ändern"
B008    DEFB 13
B009    DEFW #1621
B010    DEFB 12
B011    DEFM "3 = löschen"
B012    DEFB 13
B013    DEFW #1521
B014    DEFB 9
B015    DEFM "4 = Sort"
B016    DEFB 13
B017    DEFW #1421
B018    DEFB 10
B019    DEFM "5 = laden"
B020    DEFB 13
B021    DEFW #1321
B022    DEFB 14
B023    DEFM "6 = speichern"
B024    DEFB 14
B025    DEFB 128
```

Nach der oben genannten Befehlsfolge erscheint das Menü (zwar nicht gerade sehr einfallsreich) auf dem Bildschirm. Versuchen Sie nicht die Zeilennummer mit dem B einzugeben. Dies soll nur heißen, daß dieser Programmteil in File B gehört.

Nun muß ja noch die Selektion erfolgen.

```
A016    LOOP01 CALL GET
A017    LD HL,TAB001
A018    LD C,A
A019    CALL SUCTAB
A020    JR LOOP01
```

Diese Schleife wird solange durchlaufen bis eine gültige Eingabe erfolgt. Die Tabelle legen wir wieder in File B.

```
B026    TAB001 DEFM "1#
B027    DEFW NEU
B028    DEFM "2"
B029    DEFW AEN
B030    DEFM "3"
B031    DEFW LOE
B032    DEFM "4"
B033    DEFW SOR
B034    DEFM "5"
B035    DEFW LAD
B036    DEFM "6"
B037    DEFW SPE
B038    DEFB 128
```

Damit hätten wir schon viel erreicht. Die Sprünge werden ausgeführt, es fehlen die zugehörigen Programmteile. Fangen wir mit dem einfachen an.

```
A021    TATANZ DEFB 0
```

Diese Variable soll die Anzahl der Einträge zählen. Löschen Sie im SORT-Teil die Zeile 3950. Diese Variable wird von uns jetzt übergeben.

```
A022    SOR CALL SORT
A023    JP HAUPT
```

Bevor Sie diese Routine aufrufen, sollten Sie die beiden Slicer bestim-

```
F389 3E00      7540           LD   A,0
F38B DD211C75  7550           LD   IX,HEADER
F38F 111100    7560           LD   DE,17
F392 CD5605    7570           CALL #0556
F395 01C800    7580           LD   BC,200
F398 CD3D1F    7590           CALL PAUSE
F39B 37        7600           SCF
F39C 3EFF      7610           LD   A,255
F39E DD213075  7620           LD   IX,PLACE
F3A2 116400    7630           LD   DE,LENGTH
F3A5 CD5605    7640           CALL #0556
F3A8 C9        7650           RET
               7652 ***
               7653 *** LISTING 26
               7654 ***
F3A9 3E00      7660 SAVE_H LD   A,0
F3AB DD211C75  7670           LD   IX,HEADER
F3AF 111100    7680           LD   DE,17
F3B2 CDC204    7690           CALL #04C2
F3B5 3EFF      7700           LD   A,255
F3B7 DD213075  7710           LD   IX,PLACE
F3BB 116400    7720           LD   DE,LENGTH
F3BE CDC204    7730           CALL #04C2
F3C1 C9        7740           RET
```

men. Selbstverständlich kann das Programm dies auch abfragen. Versuchen Sie eine Eingabe zu steuern.

Nach dem Sort springt das Programm ins Menü.

Als nächstes wollen wir auch Datenfelder laden und auch wieder abspeichern (kopflos). Ersetzen Sie in Zeile 7400 den Wert 30000 durch folgenden zu berechnenden Wert: PRINT PEEK 23627+256*PEEK 23628

Mit dieser Berechnung bestimmen wir das DATAfeld. Den Wert LENGTH in Zeile 7410 legen wir mit 6009.

```
A024    LAD CALL LOAD
A025        JP HAUPT
```

Das Speichern erfolgt nach gleichem Muster:

```
A026    SPE CALL SAVE
A027        JP HAUPT
```

Der Bereich des Einholens eines Eintrags oder das Ändern und Löschen wird natürlich nicht so einfach und kurz ausfallen. Aus diesem Grund werde ich mich an dieser Stelle auf Tips beschränken.

Löschen Sie den Bildschirm und bringen Sie mit einem PRINT-Befehl 4 mal 15 Felder mit Punkten auf den Bildschirm. Setzen Sie die Bildschirmposition auf den ersten Punkt des ersten Eintrags. Suchen Sie den ersten freien Platz mit einer kleinen Schleife.

```
NEU      LD HL,(VARS)
         LD DE,8
         ADD HL,DE
LOOP11   LD A,(HL)
         CP A,(HL)
         JP Z,LEER
         CP 128
         JP Z,ENDE
         LD DE,60
         ADD HL,DE
         JR LOOP11
```

Die Stelle ENDE sollte einen Hinweis "VOLLE DATEI" auf den Bildschirm bringen, den Screen löschen und nach HAUPT springen.

Der wirklich interessante Vorgang wird der Teil LEER sein. Es wurde ein freier Eintrag gefunden.

Da die Position bereits gesetzt ist, geben Sie nur folgende Folge ein:

```
LEER     LD B,15
         PUSH HL
         CALL INPUT1
         POP HL
         LD DE,15
         ADD HL,DE
```

Das Register HL enthält ja schon die DATAadresse. Mit Ende dieses Teiles sind Sie bereit um den nächsten Adreßteil eintragen zu können. Eine falsche Eingabe sollten Sie durch ein kleines Unterprogramm abfangen. Fragen Sie zu diesem Zweck, ob die Eingabe korrekt ist oder nicht. Wenn nicht, löschen Sie diesen Eintrag mit dem Unterprogramm TEILOE und starten Sie neu.

Auf diese Art und Weise holen Sie alle 4 Adreßteile in den Speicher. Prüfen Sie nun das erste Byte des Namens auf " ". Sollte dies der Fall sein, so löschen Sie den kompletten Eintrag, er würde sowieso beim nächsten Neueintrag überschrieben. Nun erhöhen Sie noch unseren Zähler TATANZ:

```
LD DE,(TATANZ)
INC DE
LD (TATANZ),DE
```

Beim Löschen oder Ändern benötigen wir eine Suchroutine. Wir schaffen uns einen Buffer von 15 Zeichen Länge

```
Buffer DEFS 15
```

um in ihm einen Sucheintrag abspeichern zu können. Vergleichen Sie Byte für Byte die Namen. Wenn Sie gleich sind führen Sie die entsprechende Routine aus. Ansonsten erhöhen Sie den Zeiger um 60 und vergleichen weiter.

```
         LD HL,(VARS)
         LD DE,8
         ADD HL,DE
         EX DE,HL
WEITER   LD B,15
         PUSH DE
         LD HL,BUFFER
LOOP12   LD A,(DE)
         CP (HL)
         JR NZ,UNGLEI
         INC DE
         INC HL
         DJNZ LOOP12
         JR GLEICH
UNGLEICH POP DE
         LD HL,60
         ADD HL,DE
         EX DE,HL
         JR WEITER
GLEICH   POP DE
```

DE zeigt nun auf das erste Byte des gefundenen Namens. Machen Sie mit diesem Eintrag was Sie wollen. Zeigen Sie ihn an und lassen Sie ihn ändern, oder löschen Sie ihn. Setzen Sie aber TATANZ dann zurück. Eigentlich war das ja schon alles! Der Rest sind Kleinigkeiten. Zum Beispiel sollten Sie vor dem SORT etwas abfangen. Wenn kein oder nur ein Eintrag existiert, soll der Sort nicht anlaufen. Entsprechende Meldungen gehören auf den Bildschirm.

Wenn Sie die Anzahl der belegten Files auf dem Bildschirm präsent haben möchten, so weisen Sie ihr einen Platz zu. Folgende Befehlsfolge bringt die Zahl auf den Screen:

```
LD DE,TATANZ
CALL DEZIM2
```

Sie müssen eine ganze Reihe von Datenfeldern schaffen. Jede Bemerkung oder Aufforderung, Kommentar oder Fehlermeldung gehört auf den Bildschirm. (Harald Wilhelm)

Entweichen

Ein Spiel in vier Bildern mit guter 3D-Grafik und Eingriff in den Sinclair-Zeichensatz.

Ein 3D-Programm braucht viele Grafikzeichen. Der Spectrum läßt jedoch nur 21 Grafikzeichen zu. Über den Umweg der Zeichensatzänderung werden hier jedoch 42 »UDGs« definiert. Die Systemvariable in der Adresse 23606 und 23607 zeigt auf den Anfang des jeweiligen Zeichensatzes. Der zweite Zeichensatz in diesem Spiel beginnt bei der Adresse 64600. Der Original-Zeichensatz hat die Startadresse 15360 im ROM. Dieses Spiel ist also nicht nur als Spiel zu betrachten, sondern auch als Hinweis und Anleitung zum Aufbau von »SCREEN$« und zur freien Verwendung von beliebigen Zeichensätzen. Dennoch lohnt sich auch das Spiel selbst, es ist jedenfalls recht mühsam, das Ziel zu erreichen. Sie haben einem Gefängnis, einem Irrgarten, einer Brücke und einer Treppe voller Tücken zu entweichen. (H. Pallada)

HSC	Die höchste Punktzahl
LI	Die Anzahl der Leben
SC	Die Punktzahl
Schirm eins:	
NN	Der Platz, von dem der neue Stein fortgenommen werden soll
WW	Ein neuer Stein ist gepackt worden
X	Der Platz, an dem der Scheinwerfer abgedruckt werden soll
XX, YY	Der Platz, an dem die Person abgedruckt werden soll
Schirm zwei:	
A, B	Der Platz, an dem die Korridore abgeschlossen werden
BON, BONUS	Die Bonuspunkte
FB	Die Farbe der Person
SCC	Die Anzahl der aufgegessenen Punkte
XX, YY, X, Y	Der Platz, wo die Person abgedruckt werden soll
Schirm drei:	
B	Eventuelle Bewegung des Springers
PL	Der Platz, wo der neue Stein fortgenommen werden soll
QQ	Die Person trägt einen Stein
U	Eine Taste ist gedrückt worden
WW, W	Das Spiel ist verloren oder gewonnen
X, Y	Der Platz, wo die Person abgedruckt werden soll
Schirm vier:	
M	Die Höhe der Person
SCC	Die Punktzahl des vierten Schirmes
QQ	Die Person nimmt eine andere Richtung
WW	Die Person ist gesprungen
X	Der Platz des Balles

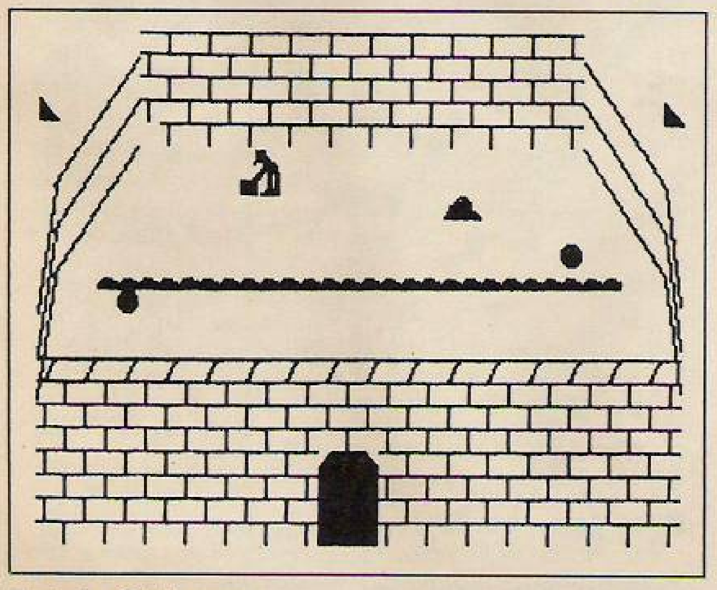
▲ **Die im Listing »Entweichen« benutzten Variablen auf einen Blick**

Struktur des Programms

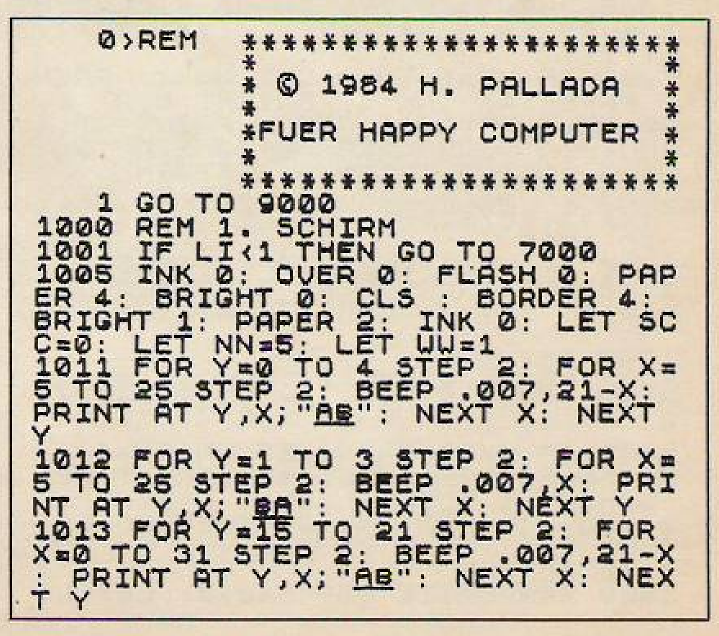
Zeilen	Funktion
1000 - 2000	Schirm eins
1005 - 1030	Erstellung des Screens
1030 - 1060	Bewegung der gelben Scheinwerfer
1070 - 1085	Absuchen der Tastatur
1086	Die Person packt einen neuen Stein, schaut,
1088	ob die Person auf dem Hinweg oder auf dem Rückweg ist
1090 - 1092	Die Person ist auf dem Hinweg
1094 - 1100	Die Person ist auf dem Rückweg
1095	Erhöhung der Punktzahl
1500 - 1520	Verlust eines Lebens
1800 - 1810	Ende des ersten Schirmes
2000 - 3000	Schirm zwei
2010 - 2200	Erstellung des Screens
2210	»POKE 23673« und »POKE 23674«: Feststellen der Zeit
2230 - 2260	Absuchen der Tastatur
2270	Ende des zweiten Schirmes
2290	Erhöhung der Punktzahl
2310	Abschluß der Korridore
2320	Änderung der Farbe der Person, wenn alle Punkte aufgegessen sind und die Person wieder beim Eingang ist
2325	Die Punktzahl wird erhöht mit dem Bonus
2335	Druckausgabe
3000 - 4000	Schirm drei
3100 - 3320	Herstellung des Schirmes
3320	Bewegung des Bootes
3330	Der weiße Leuchtblitz
3335 - 3345	Die Bewegung des Springers
3355	»QQ« ist 1, wenn die Person den Stein trägt
3365 - 3460	Bewegung der Person, wenn sie keinen Stein trägt
3460 - 3500	Bewegung der Person, wenn sie einen Stein trägt
3510	Der Springer sprüht, wenn »U« größer ist als 2 Mit den drei »POKE« wird gesprungen (zum vierten Teil der Zeile 3340)
4000 - 5000	Schirm vier
4000 - 4220	Erstellung des Screens
4230 - 4200	Bewegung der Kugeln
4260 - 4280	Absuchen der Tastatur
4300	Wenn »M« kleiner ist als Null, dann ist die Person entwichen
4320 - 4340	Ein Sprung ist gemacht worden
4370 - 4460	Der Roboter versucht, die Person wegzureißen
6000 - 7000	Die Person ist entwichen
6070	Färbung der unteren zwei (INPUT-)Zeilen mit der Adresse 23624
7000 - 8000	Ende des Spiels
7025	Eingabe des Namens. In die Adresse 23658 wird »8« »gePOKEd« für automatischen »CAPS-SHIFT«
7050	Ende des Spieles. Zurück in ROM-Zeichensatz mit den Adressen 23606 und 23607
9000 - 9999	Startposition des Programms
9000	Reserviert Raum für die Maschinensprache und für das zweite Alphabet
9005	Deklaration der Variablen
9011 - 9030	Herstellung der Zeichnung
9020	Färbung der untersten zwei (INPUT-)Zeilen mit der Adresse 23624
9050	Transport der Buchstaben von ROM nach RAM
9055	Änderung der neuen Zeichen im RAM
9065	Herstellung der Grafikzeichen
9070	Die verschiedenen Klänge werden »gePOKEd«
9085 - 9300	Erklärung des Spiels
9500 - 9700	DATA für die Maschinensprache, die Grafikzeichen und den zweiten Zeichensatz
9800	Eine Routine, die die Geschwindigkeit der Ausgabe regelt.

◀ **Diese Tabelle gibt den Programmaufbau wieder**

▼ **Basic-Listing »Entweichen« mit UDGs (unterstrichen)**



Typische Spielszene



```
   0>REM  ***********************
          *                     *
          * © 1984 H. PALLADA   *
          *                     *
          *FUER HAPPY COMPUTER  *
          *                     *
          ***********************
   1 GO TO 9000
1000 REM 1. SCHIRM
1001 IF LI<1 THEN GO TO 7000
1005 INK 0: OVER 0: FLASH 0: PAP
ER 4: BRIGHT 0: CLS : BORDER 4:
BRIGHT 1: PAPER 2: INK 0: LET SC
C=0: LET NN=5: LET WW=1
1011 FOR Y=0 TO 4 STEP 2: FOR X=
5 TO 25 STEP 2: BEEP .007,21-X:
PRINT AT Y,X;"AB": NEXT X: NEXT
Y
1012 FOR Y=1 TO 3 STEP 2: FOR X=
5 TO 25 STEP 2: BEEP .007,X: PRI
NT AT Y,X;"BA": NEXT X: NEXT Y
1013 FOR Y=15 TO 21 STEP 2: FOR
X=0 TO 31 STEP 2: BEEP .007,21-X
: PRINT AT Y,X;"AB": NEXT X: NEX
T Y
```

```
1014 FOR Y=16 TO 20 STEP 2: FOR
X=0 TO 31 STEP 2: BEEP .007,X: P
RINT AT Y,X;"BA": NEXT X: NEXT Y
1015 PAPER 1: FOR X=4 TO 10: PRI
NT AT X,0;" ";AT X,31;" ": NEXT
X: PRINT INK 5;AT 3,31;"e";AT 3,
0; INK 5;"e"
1016 BRIGHT 0: PAPER 6: FOR X=0
TO 31 STEP 2: PRINT AT 14,X;"CA"
: NEXT X
1019 FOR X=122 TO 83 STEP -15: P
LOT 6,X: DRAW 33,43: PLOT 7,X: D
RAW -7,-43: PLOT 249,X: DRAW -33
,43: PLOT 249,X: DRAW 6,-40: NEX
T X
1020 INPUT #0: PRINT #0;"SCHIRM:
1 SCORE:";SC;" LEBEN:";L$( TO LI
)
1021 BRIGHT 1: PRINT INK 3; PAPE
R 2;AT 18,14;"b■c";AT 19,14;"■■■
";AT 20,14;"■■■";AT 21,14;"■■■":
 INK 0: PLOT 124,0: DRAW 0,30
1022 FOR X=3 TO 28: BEEP .007,28
-X: PRINT INK 1; PAPER 5; BRIGHT
 1;AT 10,X;"f": NEXT X
1025 LET XX=5
1026 PRINT INK 0; PAPER 4; BRIGH
T 0;AT 7,20;"La"
1030 FOR X=4 TO 27: IF ATTR (11,
X)=33 OR ATTR (9,30-X)=33 THEN G
O TO 1500
1031 PRINT INK 6; PAPER 4;AT 11,
X;"w";AT 9,30-X;"w": GO SUB 1070
: PRINT PAPER 4;AT 11,X;" ";AT 9
,30-X;" ": NEXT X
1032 FOR X=27 TO 4 STEP -1: IF A
TTR (11,X)=33 OR ATTR (9,30-X)=3
3 THEN GO TO 1500
1033 PRINT INK 6; PAPER 4;AT 11,
X;"w";AT 9,30-X;"w": GO SUB 1070
: PRINT PAPER 4;AT 11,X;" ";AT 9
,30-X;" ": NEXT X
1060 GO TO 1029
1070 PAPER 4: BRIGHT 0: IF INKEY
$="6" THEN PRINT AT XX,10;"  ";A
T XX+1,10;"  ": LET XX=XX+1*(XX<
12): IF XX=12 AND WW=1 THEN RAND
OMIZE USR 64024: LET WW=0
1075 IF INKEY$="7" THEN PRINT AT
 XX,10;"  ";AT XX+1,10;"  ": LET
 XX=XX-1*(XX>5): IF XX=5 THEN LE
T WW=1
1080 IF INKEY$=" " THEN IF XX=8
OR XX=11 THEN RANDOMIZE USR 6412
0: PRINT AT 8+3*(XX=11),10;"  ";
AT 9+3*(XX=11),10;"  ": LET XX=X
X+3-6*(XX=11): RANDOMIZE USR 641
44: GO TO 1062

1085 PRINT INK 1; PAPER 5; BRIGH
T 1;AT 10,3;"ffffffffffffffffffff
ffffff"
1086 IF XX=5 THEN LET WW=1: PRIN
T PAPER 4; INK 1; BRIGHT 0;AT 5,
10;"HJ";AT 6,10;"IK": PRINT AT 4
,NN;" ": FOR S=20 TO 40 STEP 5:
BEEP .009,S: NEXT S: RETURN
1087 IF ATTR (XX,10)=105 OR ATTR
 (XX+1,10)=105 THEN LET WW=1: GO
 TO 1500
1088 IF WW=0 THEN GO TO 1093
1090 PRINT PAPER 4; INK 1; BRIGH
T 0;AT XX,10;"DF";AT XX+1,10;"EG
"

1092 RETURN
1094 PRINT PAPER 4; INK 1; BRIGH
T 0;AT XX,10;"F ";AT XX+1,10;"G
"

1095 IF XX=6 THEN LET NN=NN+1: L
ET SC=SC+10: INPUT #0: PRINT #0;
"SCHIRM:1 SCORE:";SC;" LEBEN:";L
$( TO LI): IF NN>=26 THEN GO TO
1800
1100 RETURN
1500 RANDOMIZE USR 64048: LET LI
=LI-1: INPUT #0: PRINT #0;"SCHIR
M:1 SCORE:";SC;" LEBEN:";L$( TO
LI): GO TO 1000
1520 GO TO 1025
1800 FOR X=0 TO 30: NEXT X: CLS
: FOR X=0 TO 30: PRINT INK 0; PA
PER 2;AT RND*20,RND*30;"BA": BEE
P .08,X: NEXT X
1810 FOR X=0 TO 100: NEXT X:: RE
M ENDE 1. SCHIRM
2000 REM 2. SCHIRM
2005 IF LI<1 THEN GO TO 7000
2010 BRIGHT 0: BORDER 3: PAPER 3
: CLS
2020 BEEP .01,10: PAPER 3: PRINT
 AT 0,0; INK 6;"d"; INK 1;"■■■■
■■■■■■■■■■■■■■■■■■■■■■"
2025 BEEP .01,11: PRINT INK 6; P
APER 5;"d"; INK 6;"■■■■■■■■■■
■■■■■"; INK 5; PAPER 6;"d"; IN
K 1;"■"; PAPER 5; INK 6;"■■■■■
g"; INK 1;"■"
2030 BEEP .01,12: PRINT INK 7;".
...................."; INK 5;"■"
; INK 1;"■"; INK 7;"......."; IN
K 5;"■"; INK 1;"■"
2035 BEEP .01,13: PRINT INK 5;"d
"; INK 1;"■■■■■■"; INK 7;"."; I
NK 5;"d"; INK 1;"■"; INK 7;".";
INK 5;"d"; INK 1;"■"; INK 7;".";
 INK 5;"d"; INK 1;"■"; INK 7;".
"; INK 5;"d"; INK 1;"■"; INK 7;"
."; INK 5;"■"; INK 1;"■"; INK 7;"
."; INK 5;"d"; INK 1;"■■■■"; INK
 7; "."; INK 5;"■"; INK 1;"■"
2040 BEEP .01,14: PRINT INK 5;"■
"; INK 1;"■"; INK 6;"■■■■"; INK
6; PAPER 5;"g"; PAPER 3; INK 1;"
■"; INK 7;"."; INK 5;"■"; INK 1;
"■"; INK 7;"."; INK 5;"■"; INK 1
;"■"; INK 7;"."; INK 5;"■"; INK
1;"■"; INK 7;"."; INK 5;"■"; INK
 1;"■"; INK 7;"."; INK 5;"■"; IN
K 1;"■"; INK 7;"."; PAPER 5; INK
 6;"d"; PAPER 3; INK 6;"■"; INK
6;"■■g"; INK 7; "."; INK 5;"■";
INK 1;"■"
2045 BEEP .01,15: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"...."; INK
5;"■"; INK 1;"■"; INK 7;"."; INK
 5;"■"; INK 1;"■"; INK 7;"."; IN
K 5;"■"; INK 1;"■"; INK 7;"."; I
NK 5;"■"; INK 1;"■"; INK 7;".";
INK 5;"■"; INK 1;"■"; INK 7;".";
 INK 5;"■"; INK 1;"■"; INK 7;"..
....."; INK 5;"■"; INK 1;"■"
2050 BEEP .01,16: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
d"; INK 1;"■"; INK 7;"."; INK 5;
"■"; INK 1;"■"; INK 7;"."; INK 5
;"■"; INK 1;"■"; INK 7;"."; INK
5;"■"; INK 1;"■"; INK 7;"."; INK
 5; PAPER 6;"g"; INK 6; PAPER 3;
"g"; PAPER 3; INK 7;"."; INK 5;"
■"; INK 1;"■"; INK 7;"."; INK 5;
"■"; INK 1;"■"; INK 7;"."; INK 5
;"d"; INK 1;"■"; INK 7;"."; INK
5;"d"; INK 1;"■■■■"
2060 BEEP .01,17: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
■"; INK 1;"■"; INK 7;"."; INK 6;
 PAPER 5;"d"; PAPER 3; INK 6;"g"
; INK 7;"."; INK 5;"■"; INK 1;"■
"; INK 7;"."; INK 5;"■"; INK 1;"
■"; INK 7;"...."; INK 5;"■"; INK
 1;"■"; INK 7;"."; INK 5;"■"; IN
K 1;"■"; INK 7;"."; INK 5;"■"; I
NK 1;"■"; INK 7;"."; INK 6; PAPE
R 5;"d■■g"; INK 1;"■"
2065 BEEP .01,18: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
■"; INK 1;"■"; INK 7;"...."; INK
 5;"■"; INK 1;"■"; INK 7;"."; IN
```

Basic-Listing »Entweichen« (Fortsetzung)



Hallo Computer-Freaks aufgepaßt:

Deutsche Abenteuerspiele der Spitzenklasse – in Stil und Aufmachung mit amerikanischen Adventure ebenbürtig – für den Commodore 64.

Zum Super-Sparpreis von DM 34,90* (sFr. 29,50 öS 314,10) für 2 Top-Abenteuerspiele

(2 doppelseitig bespielte Disks in 1 Abenteuer-Paket)

Abenteuer-Paket 1

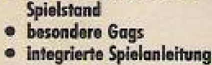
Sagor der Eroberer



Ein Abenteuerspiel mit aufwendiger Hires-Grafik, das sich über drei Diskettenseiten erstreckt. Finden Sie den Ring des Schlangengottes »AMON« und befreien Sie mit ihm die Welt von allem Bösen!

Sagor bietet Ihnen:

- 27 Hires-Bilder
- variablen Spielverlauf
- Musik
- lad- und speicherbarer Spielstand
- besondere Gags
- integrierte Spielanleitung



Operation Neptun



Alarm in der Kommandozentrale Ihres U-Boots: »Defekter Satellit stürzt mit Atomreaktor in die Untiefen des Meeres.« Ihnen bleibt nicht viel Zeit, die Erde vor der radioaktiven Verseuchung zu retten.

Operation Neptun bietet Ihnen:

- 40 Hires-Bilder
- lad- und speicherbarer Spielstand
- Help-Funktion
- integrierte Spielanleitung

Zusammen nur DM 36,90* (sFr. 29,50/öS 314,10)
Best.-Nr. MD 245 A

Abenteuer-Paket 2

Drachental

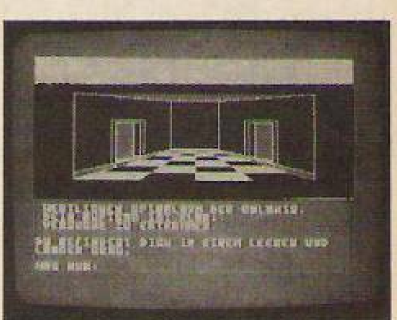


Werden Sie zum mittelalterlichen Helden. Große Taten sind zu vollbringen. Besiegen Sie die »bösen« Drachen. Am Ende erwartet Sie die schöne Prinzessin.

Drachental bietet Ihnen:

- sehr schöne Hires-Grafik; teilweise mit Zeichentrickeffekt
- integrierte Musik
- variablen Spielverlauf
- lad- und speicherbarer Spielstand
- integrierte Spielanleitung

Flucht ins Paradies



Abenteuer im Jahre 2293. Sie sitzen in einer kleinen Sternenbasis – einer völlig aus der Mode gekommenen, drittklassigen Galaxie – fest. Starbase III – das Paradies aller Galaxien – lockt. Finden Sie den Weg in Ihr Paradies.

Flucht ins Paradies bietet Ihnen:

- Hires-Multicolor-Grafik
- 50 Bilder
- sehr großer Wortschatz
- lad- und speicherbarer Spielstand
- integrierte Spielanleitung

Zusammen nur DM 34,90* (sFr. 29,50/öS 314,10)
Best.-Nr. MD 246 A

Bitte verwenden Sie für diese Software-Bestellung die eingeheftete Postscheck-Zahlkarte.

Am besten gleich bestellen!

Markt&Technik
Verlag Aktiengesellschaft
Buchverlag

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt&Technik-Vertriebs AG, Kollerstraße 3, CH-6300 Zug, ☎ 042/22 31 55
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien ☎ 02 22/67 75 26

MD = Diskette
A = Commodore
* inkl. MwSt. Unverbindliche Preisempfehlung.

```
K 5;"■"; INK 1;"■"; INK 7;"."; I
NK 5;"d"; INK 1;"■■■■"; INK 7;".
"; INK 5;"■"; INK 1;"■"; INK 7;"
."; INK 5;"■"; INK 1;"■"; INK 7;
"...."; INK 5;"■"; INK 1;"■"
2070 BEEP .01,19: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
■"; INK 1;"■"; INK 1;"■■■■■"; I
NK 7;"."; INK 5; PAPER 6;"g"; PA
PER 3; INK 6;"g"; INK 7;"."; INK
 5; PAPER 6;"g"; PAPER 3; INK 6;
"■"; INK 6;"■g"; INK 7;"."; INK
 5; PAPER 6;"g"; PAPER 3; INK 6;
"g"; INK 7;"."; INK 5;"■"; INK 1
;"■"; INK 1;"■■■"; INK 7;"."; IN
K 5;"■"; INK 1;"■"
2075 BEEP .01,20: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 6;
PAPER 5;"d"; PAPER 3; INK 6;"■■
■■g"; INK 7;"............"; IN
K 6; PAPER 5;"d"; PAPER 3; INK 6
;"■■■g"; INK 7;"."; INK 5;"■"; I
NK 1;"■"
2080 BEEP .01,21: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"........."
; INK 5;"d"; INK 1;"■"; INK 7;".
"; INK 5;"d"; INK 1;"■"; INK 7;"
."; INK 5;"d"; INK 1;"■"; INK 1;
"■■■"; INK 7;"......."; INK 5;"■
"; INK 1;"■"
2085 BEEP .01,22: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
d"; INK 1;"■■■■■■"; INK 7;".";
INK 5;"■"; INK 1;"■"; INK 7;".";
 INK 5;"■"; INK 1;"■"; INK 7;".
; INK 5; PAPER 6;"g"; PAPER 3; I
NK 6;"■■■g"; INK 7;"."; INK 5;"d
"; INK 1;"■"; INK 7;"."; INK 5;"
d"; INK 1;"■"; INK 7;"."; INK 5;
"■"; INK 1;"■"
2090 BEEP .01,23: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 6;
PAPER 5;"d"; INK 6;"■■■■g"; PAP
ER 3; INK 1;"■"; INK 7;"."; INK
5;"■"; INK 1;"■"; INK 7;"."; INK
 5; PAPER 6;"g"; PAPER 3; INK 6;
"g"; INK 7;"......"; INK 5;"■";
 INK 1;"■"; INK 7;"."; INK 5;"■"
; INK 1;"■"; INK 7;"."; INK 5;"■
"; INK 1;"■"
2095 BEEP .01,24: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"......."; P
APER 6; INK 5;"g"; PAPER 3; INK
6;"g"; PAPER 3; INK 7;"."; INK 5
; PAPER 6;"g"; PAPER 3; INK 6;"g
"; INK 7;"....."; INK 5;"d"; INK
 1;"■"; INK 7;"..."; INK 5;"■";
INK 1;"■"; INK 7;"."; INK 5;"■";
 INK 1;"■"; INK 7;"."; INK 5;"■"
; INK 1;"■"
2100 BEEP .01,25: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
d"; INK 1;"■"; INK 7;"."; INK 5;
"d"; INK 1;"■"; INK 7;"........
; INK 5;"d"; INK 1;"■"; INK 7;".
"; INK 5;"■"; INK 1;"■"; INK 7;"
."; INK 5;"d"; INK 1;"■■■■"; INK
7;"."; INK 5;"■"; INK 1;"■"; INK
 7;"."; INK 5;"■"; INK 1;"■"
2105 BEEP .01,26: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 5;"
■"; INK 1;"■"; INK 7;"."; INK 5;
"■"; INK 1;"■"; INK 7;"."; INK 5
;"d"; INK 1;"■■■■"; INK 1;"■"; I
NK 7;"."; INK 5;"■"; INK 1;"■";
INK 7;"."; INK 5;"■"; INK 1;"■";
 INK 7;"."; INK 6; PAPER 5;"d■g"
; PAPER 3; INK 1;"■"; INK 7;".";
 INK 5;"■"; INK 1;"■"; INK 7;".
; INK 5;"■"; INK 1;"■"
2110 BEEP .01,27: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"."; INK 6;
PAPER 5;"d"; PAPER 3; INK 6;"g";
 INK 7;"."; INK 5;"■"; INK 1;"■"
; INK 7;"."; INK 6; PAPER 5;"d";
 PAPER 3; INK 6;"■■■■g"; INK 7;"
."; INK 5;"■"; INK 1;"■"; INK 7;
"."; INK 5;"■"; INK 1;"■"; INK 7
;"..."; INK 5;"■"; INK 1;"■"; IN
K 7;"."; INK 6; PAPER 5;"d"; INK
 6; PAPER 3;"g"; INK 7;"."; INK
6; PAPER 5;"d"; PAPER 3; INK 6;"
g"
2115 BEEP .01,27: PRINT INK 5;"■
"; INK 1;"■"; INK 7;"...."; INK
5;"■"; INK 1;"■"; INK 7;"......
."; INK 5;"■"; INK 1;"■"; INK 7;
"."; INK 5;"■"; INK 1;"■"; INK 7
;"..."; INK 5;"■"; INK 1;"■"; IN
K 7;"...."
2120 BEEP .01,28: PRINT INK 5;"■
"; INK 1;"■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■"
2125 BEEP .01,29: PRINT INK 6; P
APER 5;"d"; INK 6; PAPER 3;"■■■■
■■■■■■■■■■■■■■■■■■■■■■■■■g"
2200 INK 7: PAPER 3: BRIGHT 0
2210 POKE 23674,0: POKE 23673,24
0: LET BON=255: LET SCC=-1: LET
FB=7: LET XX=2: LET YY=0: LET X=
2: LET Y=0: FOR S=0 TO 10: NEXT
S
2220 GO TO 2300
2230 LET A$=INKEY$
2235 IF A$="5" THEN LET Y=YY-1+(
YY<0 OR ATTR (XX,YY-1)<>31)-(ATT
R (XX,YY-1)=68): IF YY<>Y THEN P
RINT PAPER 3;AT X,YY;" ": IF ATT
R (XX,YY-1)=68 THEN LET LI=LI-1:
 RANDOMIZE USR 64048: GO TO 2000
2240 IF A$="8" THEN LET Y=YY+1-(
YY>30 OR ATTR (XX,YY+1)<>31)+(AT
TR (XX,YY+1)=68): IF Y<>YY THEN
PRINT PAPER 3;AT X,YY;" ": IF AT
TR (XX,YY+1)=68 THEN LET LI=LI-1
: RANDOMIZE USR 64048: GO TO 200
0
2250 IF A$="6" THEN LET X=XX+1-(
XX>18 OR ATTR (XX+1,YY)<>31)+(AT
TR (XX+1,YY)=68): IF X<>XX THEN
PRINT PAPER 3;AT XX,Y;" ": IF AT
TR (XX+1,YY)=68 THEN LET LI=LI-1
: RANDOMIZE USR 64048: GO TO 200
0
2260 IF A$="7" THEN LET X=XX-1+(
XX<2 OR ATTR (XX-1,YY)<>31)-(ATT
R (XX-1,YY)=68): IF X<>XX THEN P
RINT PAPER 3;AT XX,Y;" ": IF ATT
R (XX-1,YY)=68 THEN LET LI=LI-1:
 RANDOMIZE USR 64048: GO TO 2000
2270 LET XX=X: LET YY=Y: IF YY=3
0 AND FB=0 THEN RANDOMIZE USR 64
072: RANDOMIZE USR 64072: PAUSE
100: GO TO 3000
2280 IF YY=31 THEN LET YY=30: LE
T Y=30
2300 IF SCREEN$ (X,Y)="." THEN L
ET SCC=SCC+1: LET SC=SC+.5: BEEP
 .005,X+Y: INPUT #0: PRINT #0;"S
CHIRM:2 SCORE:";INT (SC);" LEBEN
:";L$( TO LI): LET BONUS=10*(BON
-PEEK 23673)*(PEEK 23674=0): PRI
NT INK 7;AT 21,0;"BONUS:";AT 21,
6;"    ";AT 21,6;BONUS
2310 IF SCC=220 THEN IF RND>.65
THEN LET A=1+INT (RND*18): LET B
=1+(RND*29): IF ATTR (A,B)=31 AN
D A<>2 AND A<>11 AND B<26 THEN P
RINT ; INK 4; PAPER 0; BRIGHT 1;
AT A,B;"■"
2320 IF SCC=220 THEN IF XX=2 AND
 YY=0 AND FB<>0 THEN LET FB=0: B
EEP .009,0: BEEP .007,30: BEEP .
008,10
2325 IF SCC=220 THEN IF BONUS<>0
 THEN LET SC=SC+BONUS: RANDOMIZE
 USR 64120: INPUT #0: PRINT #0;;
```

Basic-Listing »Entweichen« (Fortsetzung)

```
"SCHIRM:2 SCORE:";INT (SC);" LEB
EN:";L$( TO LI): LET BONUS=0
2335 PRINT INK FB; BRIGHT 1;AT X
.Y;"i": GO TO 2230
3100 LET SC=INT (SC)
3105 LET SCC=0: LET WW=0: LET U=
0: LET X=3: LET Y=14: BRIGHT 0:
INK 7: PAPER 0: BORDER 0: CLS :
INPUT #0: PRINT #0;"SCHIRM:3 SCO
RE:";INT SC;" LEBEN:";L$( TO LI)
3110 FOR X=5 TO 15: PRINT INK 2;
AT X,0;"██████";AT X,20;"██████
██": BEEP .01,2*X: NEXT X
3120 FOR X=0 TO 2: PRINT INK 4;A
T 4-X,X;"d": BEEP .02,3*X: NEXT
X
3130 FOR X=0 TO 2: PRINT INK 4;A
T 4-X,20+X;"d": BEEP .01,6+3*X:
NEXT X
3135 FOR X=0 TO 2: FOR Y=1 TO 5:
 PRINT INK 4;AT 4-X,X+Y;"█": BEE
P .01,12+3*X: NEXT Y: NEXT X
3140 FOR X=0 TO 2: FOR Y=1 TO 8:
 PRINT INK 4;AT 4-X,20+X+Y;"█":
NEXT Y: BEEP .01,18+3*X: NEXT X
3145 FOR X=7 TO 9: PRINT INK 4;
PAPER 3;AT X-5,15-X;"g": BEEP .0
1,3*X: NEXT X
3150 FOR X=7 TO 9: PRINT INK 4;
PAPER 3;AT X-5,38-X;"g": BEEP .0
1,(50-3*X): NEXT X
3155 FOR X=0 TO 9: FOR Y=1 TO 3:
 PRINT INK 3;AT 6+X-Y,5+Y;"█": B
EEP .01,(X+Y): NEXT Y: NEXT X
3160 FOR X=0 TO 9: FOR Y=1 TO 3:
 PRINT INK 3;AT 6+X-Y,28+Y;"█":
BEEP .01,(Y+X): NEXT Y: NEXT X
3165 FOR X=18 TO 20: PRINT INK 3
; PAPER 1;AT X-5,26-X;"g": NEXT
X
3170 FOR X=18 TO 20: PRINT INK 3
; PAPER 0;AT X-5,49-X;"g": NEXT
X
3175 FOR X=20 TO 18 STEP -1: FOR
 Y=2 TO X-6: PRINT INK 1;AT X-5,
25-X+Y;"█": BEEP .01,50-X-Y: NEX
T Y: NEXT X
3180 PRINT INK 6; PAPER 4;AT 3,7
;"d█";AT 3,20;"█g"
3185 PRINT INK 5; PAPER 4;AT 4,7
;"███";AT 4,20;"g"
3190 FOR X=16 TO 21: PRINT INK 1
; PAPER 0;AT X,21-X;"d██████████
███": BEEP .01,50-X: NEXT X
3200 FOR X=16 TO 21: PRINT INK 1
; PAPER 0;AT X,35-X;"g": BEEP .0
1,X: NEXT X
3210 PRINT PAPER 6; INK 1;AT 16,
22;"tttttt";AT 14,22;"ttttt";AT
17,22;"uuuuuu";AT 15,22;"uuuuu"
3300 LET W=0: LET PL=1: LET QQ=0
: LET X=3: LET Y=27
3305 PLOT 0,175: DRAW 255,0
3310 PRINT INK 1; PAPER 4;AT X,Y
;"i"
3320 IF SCC=110 THEN PRINT PAPER
 5; INK 0;AT 4,13;"y": FOR B=21
TO 13 STEP -1: PRINT INK 2; PAPE
R 1;AT B,26-B;"x": PAUSE 10: PRI
NT PAPER 1;AT B,26-B;" ": IF WW=
2 THEN NEXT J: LET WW=3: IF (B<>
13 OR Y<>13) THEN LET LI=LI-1: F
OR S=0 TO 3: RANDOMIZE USR 64048
: NEXT S: LET WW=2: GO TO 3370
3330 IF SCC=110 THEN IF B=20 THE
N BEEP .009,0: BORDER 7: FOR T=0
 TO 10: NEXT T: BORDER 0: BEEP .
009,20
3335 IF SCC<>110 THEN FOR B=8 TO
 30: PRINT PAPER 0; INK 7;AT 0,B
;"Ax"
3340 IF SCC<>110 THEN FOR U=1 TO
 2: GO TO 3350: LET U=1: PRINT A
T 1,B+1;"v";AT 2-(B<9 OR B>19),B
+1;"v": IF ATTR (3,B+1)=49 OR AT
TR (3,B+1)=33 THEN LET LI=LI-1:
LET WW=2: RANDOMIZE USR 64096: G
O TO 3370
3345 IF U=1 THEN PRINT PAPER 0;A
T 1,B+1;" ";AT 2-(B<9 OR B>19),B
+1;" ": LET U=0
3350 IF QQ=1 THEN GO TO 3460
3360 LET A$=INKEY$
3365 IF A$="4" AND (Y<20 AND Y>8
) THEN PRINT INK 6;AT X,Y;"█": R
ANDOMIZE USR 64192: LET WW=2: FO
R J=6 TO 12: PRINT AT J-1,Y;" ";
 INK 7;AT J,Y;"i": FOR T=1 TO 5:
 NEXT T: NEXT B: NEXT J: LET T=0
: LET WW=3: IF (B<>13 OR Y<>13)
THEN LET LI=LI-1: LET WW=2: RAND
OMIZE USR 64048
3367 IF W=4 THEN LET LI=LI-1: LE
T WW=2: PAUSE 50: LET W=0: RANDO
MIZE USR 64048
3370 IF WW=3 THEN RANDOMIZE USR
64072: RANDOMIZE USR 64072: PAUS
E 30: GO TO 4000
3372 IF LI<1 THEN GO TO 7000
3375 IF WW=2 THEN PAUSE 100: CLS
 : GO TO 3105
3380 IF A$="8" THEN IF (Y>21 OR
X<>4) THEN LET Y=Y+1-(Y>27): IF
Y<29 THEN PRINT PAPER 4+2*(Y<22
AND Y>7); INK 1;AT X,Y;"i"; INK
3;AT X,Y-1;" ": IF Y=8 OR Y=22 T
HEN PRINT AT X,Y-1; INK 4+2*(Y=2
2); PAPER 6-2*(Y=22);"g"
3385 IF A$="5" THEN IF ATTR (X,Y
-1)=7 THEN LET A$="4": LET W=4:
GO TO 3365
3390 IF A$="5" THEN IF (Y>=22 OR
 X=3) THEN LET Y=Y-1+(Y<4): IF Y
>2 THEN PRINT PAPER 4+2*(Y<21 AN
D Y>6); INK 1;AT X,Y;"i"; INK 3;
AT X,Y+1;" ": IF Y=6 OR Y=20 THE
N PRINT AT X,Y+1; INK 4+2*(Y=20)
; PAPER 6-2*(Y=20);"g"
3400 IF A$="7" THEN IF Y<6 OR Y>
20 THEN LET X=X-1+(X<4): IF X>2
THEN PRINT PAPER 4;AT X+1-(X=1),
y;" "; INK 1; PAPER 4;AT X,Y;"i"
3410 IF A$="6" THEN IF Y<6 OR Y>
20 THEN LET X=X+1-(X>3): IF X<5
THEN PRINT PAPER 4;AT X-1+(X=5),
Y;" "; INK 1; PAPER 4;AT X,Y;"i"
3420 LET A$=INKEY$
3431 IF A$="9" THEN IF X=4 THEN
IF ATTR (15,Y)=49 THEN LET PL=15
3440 IF A$="9" THEN IF X=4 THEN
IF (ATTR (17,Y)=49 OR PL=15) THE
N FOR K=5 TO (15-(PL=15)): BEEP
.003,3*K: PRINT PAPER 2; INK 7;A
T K,Y;"|": NEXT K: FOR K=(15-(PL
=15)) TO 5 STEP -1: BEEP .003,3*
K: PRINT PAPER 6; INK 0;AT K,Y;"
t";AT K+1,Y;"u"; PAPER 2-2*(K>13
);AT K+2-(K>13 AND PL=15),Y;" ":
 NEXT K: PRINT PAPER 2;AT K+1,Y;
" ";AT K+2,Y;" ": PRINT INK 0; P
APER 6;AT K,Y;"z"; PAPER 4; INK
1;"i": LET QQ=1: LET PL=17
3460 IF QQ=0 THEN GO TO 3510
3470 LET A$=INKEY$
3480 IF A$="5" THEN IF X=3 AND A
TTR (3,Y-2)=7 THEN PRINT INK 6;A
T X,Y-2;"███"; INK 5;AT X+1,Y-2;"
█": LET QQ=0: RANDOMIZE USR 6424
0: LET SCC=SCC+10: LET SC=SC+10:
 INPUT #0: PRINT #0;"SCHIRM:3 SC
ORE:";INT SC;"LEBEN:";L$( TO LI)
: GO TO 3510
3490 IF A$="5" THEN IF ATTR (X,Y
-2)<>7 THEN IF (Y>=23 OR X=3) TH
EN LET Y=Y-1+(Y<4): IF Y>2 THEN
PRINT INK 0; PAPER 6;AT X,Y-1;"z
"; PAPER 4+2*(Y<21 AND Y>6); INK
```

Basic-Listing »Entweichen« (Fortsetzung)

```
1;"i"; INK 4+2*(Y<21)-2*(Y=19);
AT X,Y+1;" d": IF Y=6 OR Y=20 TH
EN PRINT AT X,Y+1; INK 4+2*(Y=20
); PAPER 6-2*(Y=20);"g "
3495 IF A$="8" THEN IF (Y>21 OR
X<>4) THEN LET Y=Y+1-(Y>25): IF
Y<29 THEN PRINT PAPER 4+2*(Y<22
AND Y>7); INK 1;AT X,Y;"i"; PAPE
R 6; INK 0;"z"; PAPER 4+2*(Y<23
AND Y>7); INK 4+2*(Y<23)+2*(Y=23
 AND X=3);AT X,Y-2;"g ": IF Y=6
OR Y=22 THEN PRINT AT X,Y-1; INK
 4+2*(Y=22); PAPER 6-2*(Y=22);"g
"
3500 IF A$="7" THEN IF Y<6 OR Y>
20 THEN LET X=X-1+(X<4): IF X>2
THEN PRINT PAPER 4;AT X+1-(X=1),
Y;" "; INK 0; PAPER 6;AT X,Y;"z"
; PAPER 4; INK 1;"i"; PAPER 4;AT
 X+1,Y-1;"  ": IF Y=21 THEN PRI
NT INK 5; PAPER 4;AT X+1,20;"g"
3510 IF U<>0 THEN NEXT U: POKE 2
3619,13: POKE 23618,12: POKE 236
20,4
3520 NEXT B: PRINT PAPER 0;AT 0,
31;"A": GO TO 3320
4040 IF LI<1 THEN GO TO 7000
4045 BRIGHT 0: BORDER 1: LET SCC
=0: LET NN=5: PAPER 1: CLS : LET
 C$="o": LET D$="p": LET QQ=0: L
ET WW=0: LET M=19: INK 6: PAPER
2: INPUT #0: PRINT #0;"SCHIRM:4
SCORE:";SC;" LEBEN:";L$( TO LI)
4050 PRINT AT 1,1;"dj"
4060 PRINT "dgdj"
4070 PRINT "█dgdj"
4080 FOR X=1 TO 15
4090 BEEP .01,2*X: PRINT AT X+3,
X;"█dgdj"
4100 NEXT X
4110 PRINT AT 19,16;"█dgdj"
4120 PRINT AT 20,17;"█dg"
4130 PRINT AT 21,18;"█"
4140 PRINT PAPER 1; INK 6;AT 1,1
;"d";AT 2,0;"d";AT 20,19;"g"
4160 PRINT INK 2; BRIGHT 1;AT 21
,19;"████████████"; PAPER 6; BRIGHT
 0; INK 4;AT 20,19;"d███████████";
 PAPER 6;AT 19,20;"d███████████"
4170 PRINT PAPER 2; INK 4;AT 19,
29;"g";AT 20,28;"g": PRINT INK 2
; PAPER 1;AT 20,29;"g";AT 21,28;
"g"
4180 INK 3
4190 LET H$="████████████████████
████████████████"
4200 FOR Y=3 TO 21: PRINT AT Y,0
;H$( TO Y-3): BEEP .01,30-2*Y
4210 NEXT Y
4215 PRINT INK 3; PAPER 1;AT 0,1
;"k";AT 1,0;"rs"
4220 FOR X=0 TO 18: PRINT BRIGHT
 1; INK 7;AT X,25;"█": BEEP .01,
(50-2*X): NEXT X
4230 FOR X=2 TO 19: IF ATTR (X,X
)=48 THEN LET LI=LI-1: RANDOMIZE
 USR 64048: PAUSE 50: GO TO 4040
4240 PRINT INK 0; BRIGHT 1; PAPE
R 6;AT X,X;"w": PAUSE 5: PRINT A
T X,X; PAPER 2; INK 6;"d"
4250 IF WW=1 THEN NEXT S: LET WW
=0: PRINT INK 1;AT M-5,M;"█";AT
M-4,M;"█": PRINT PAPER 6; INK 0;
AT M,M;C$;AT M+1,M;D$
4260 IF INKEY$="6" THEN IF M<19
THEN LET M=M+1: PRINT INK 6; PAP
ER 2;AT M-1,M-1;"d";AT M,M-1;"g"
: LET QQ=6: IF M=1 THEN PRINT IN
K 1;AT 0,0;"█";AT 1,0;"█"
4270 IF QQ=6 THEN LET C$="k": LE
T D$="l": LET QQ=0: GO TO 4300
4275 IF INKEY$="7" THEN IF M>0 T
HEN PRINT INK 6; PAPER 2;AT M,M;
"d";AT M+1,M;"g": LET M=M-1: LET
 C$="o": LET D$="p": GO TO 4300
4280 IF INKEY$=" " THEN GO TO 43
10
4300 IF M<=1 THEN PAUSE 50: GO T
O 6000: REM ENDE SPIEL
4305 PRINT INK 2; PAPER 6; INK 0
;AT M,M;C$;AT M+1,M;D$: IF SCC<1
9-M THEN LET SC=SC+10*(19-M)-10*
SCC: LET SCC=19-M: INPUT #0: PRI
NT #0;"SCHIRM:4 SCORE:";SC;" LEB
EN:";L$( TO LI)
4307 GO TO 4350
4315 IF M<=3 THEN NEXT X
4320 PRINT PAPER 2; INK 6;AT M,M
;"d";AT M+1,M;"g"
4330 PRINT INK 0; PAPER 6; FLASH
 1; BRIGHT 1;AT M-5,M;C$;AT M-4,
M;D$: LET WW=1
4340 FOR S=20 TO 50 STEP 4: BEEP
 .03,S: NEXT X: GO TO 4230
4350 IF M<18 THEN PRINT INK 7; B
RIGHT 1;AT M-1,25;"█";AT M+1,25;
"█"; BRIGHT 1; PAPER 7; INK 3;AT
 M,25;"m"
4360 IF RND>.12 OR M>18 OR M<4 T
HEN NEXT X: GO TO 4230
4370 RANDOMIZE USR 64168:: BORDE
R 4: FOR Y=1 TO 16: IF INKEY$="
" THEN BORDER 1: LET Y=RND: GO T
O 4315
4380 NEXT Y: BORDER 1
4390 FOR Y=M TO 0 STEP -1: PRINT
 BRIGHT 1; PAPER 7;AT Y+1,25;" "
; INK 3;AT Y,25;"m": NEXT Y
4400 FOR Y=25 TO M STEP -1: PRIN
T AT 0,Y+1; PAPER 1;" "; INK 7;A
T 0,Y;"n": NEXT Y
4410 FOR Y=1 TO M-2: PRINT PAPER
 1;AT Y-1,M;" "; INK 7;AT Y,M;"n
": NEXT Y
4420 RANDOMIZE USR 64216: PRINT
PAPER 2; INK 6;AT M,M;"d";AT M+1
,M;"g"
4430 FOR Y=M-3 TO 0 STEP -1: PRI
NT PAPER 1;AT Y+2-1*(Y=M-3),M;"
";AT Y+1,M;" "; INK 7;AT Y,M;"o"
;AT Y+1,M;"p": NEXT Y: PRINT PAP
ER 1;AT Y+1,M;" ";AT Y+2,M;" "
4440 FOR Y=M+4 TO 24: PRINT AT 0
,Y-1; PAPER 1;" ";AT 1,Y-1;" ";
INK 7;AT 0,y;"o";AT 1,Y;"p": NEX
T Y: PRINT PAPER 1;AT 0,Y-1;" ";
AT 1,Y-1;" "
4450 FOR Y=22-M TO 17: PRINT AT
Y,24; PAPER 1;" ";AT Y-1,24;" ";
 INK 7;AT Y,24;"o";AT Y+1,24;"p"
: NEXT Y: PRINT PAPER 1;AT Y,24;
" ";AT Y-1,24;" "
4460 LET M=19: GO TO 4230
6005 BRIGHT 0: PAPER 6: BORDER 6
: CLS : INK 1: PRINT '"IHR AUFTR
AG IST ERFOLLT:"'"SIE SIND ENTWE
ICHT ! ! !"
6010 FOR S=-15 TO 40 STEP 5: BEE
P RND*RND,S: NEXT S
6025 PRINT '"IHRE PUNKTEZAHL IST
: ";SC
6035 IF HSC>0 THEN PRINT '"DER
BESTE SPIELER VON HEUTE"'"IST: "
; FLASH 1; INK 2; PAPER 7;B$
6040 IF HSC>0 THEN PRINT B$;" HA
T ";HSC;" PUNKTEN."
6050 PRINT '"IHRE FLUCHT HAT IHN
EN ";3-LI;" LEBEN   GEKOSTET."'"
SIE HABEN ALSO NOCH ";LI;" LEBEN
"'"QBRIG."
6060 PRINT '"MIT DIESEM ";LI;" L
EBEN KNNNEN SIE   NOCH EINE ENTW
EICHUNG VERSUCHEN."
6070 POKE 23624,56: PRINT #0;"DR
QCKEN SIE ""ENTER"" EIN FQR EINE
NEUEN ENTWEICHUNG": PAUSE 0: RAN
DOMIZE USR 64120: GO TO 1000
7000 REM ENDE DES SPIELES
```

Basic-Listing »Entweichen« (Fortsetzung)

```
7005 INK 5: PAPER 1: BRIGHT 0: B
ORDER 1: CLS : PRINT "ES TUT MIR
 LEID: SIE HABEN DREI LEBEN VERS
PIELT ! ! !"
7010 RANDOMIZE USR 64048: RANDOM
IZE USR 64048
7015 PRINT '"SIE HATTEN NUR DREI
 LEBEN, DAS  SPIEL IST ALSO ZUM
ENDE."
7020 PRINT '"SIE HABEN EINEN ""S
CORE"" ERREICHTVON: ";SC;" PONKT
E."
7025 POKE 23658,8: IF SC>HSC THE
N LET HSC=SC: PRINT '"MIT DIESEM
 ""SCORE"" SIND SIE DER BESTEN S
PIELER VON HEUTE ! ! !"'"GEBEN S
IE BITTE IHREN NAME.": INPUT B$
7030 IF HSC>0 THEN PRINT '"HEUT
E'S BESTER SPIELER: "; FLASH 1;
INK 2; PAPER 7;B$
7035 PRINT B$;" HAT ";HSC;" PONK
TE ERREICHT."
7040 INK 6: PRINT ''"DROCKEN SIE
 BITTE ""J"" EIN WENN  SIE NOCH
EIN SPIEL MNCHTEN       SPIELEN."
: INPUT J$: POKE 23658,0: IF J$=
"J" THEN LET SC=0: LET LI=3: GO
TO 1000
7050 POKE 23607,60: POKE 23606,0
: REM ZURUECK NACH NORMALEN BUCH
STABEN.
7060 STOP
9000 CLEAR 63999
9005 LET B$="/": LET HSC=0: LET
L$="iii": LET SC=0: LET LI=3
9010 FLASH 0: BRIGHT 1: OVER 1:
INK 6: PAPER 1: BORDER 1: CLS :
FOR X=0 TO 95: PLOT 125,80: DRAW
 125,X: NEXT X: FOR X=125 TO 0 S
TEP -1: PLOT 125,80: DRAW X,95:
NEXT X
9015 FOR X=0 TO 125: PLOT 125,80
: DRAW -X,95: NEXT X: FOR X=95 T
O 0 STEP -1: PLOT 125,80: DRAW -
125,X: NEXT X
9020 FOR X=0 TO 80: PLOT 125,80:
 DRAW -125,-X: NEXT X: FOR X=125
 TO 0 STEP -1: PLOT 125,80: DRAW
 -X,-80: NEXT X
9025 FOR X=0 TO 125: PLOT 125,80
: DRAW X,-80: NEXT X: FOR X=80 T
O 0 STEP -1: PLOT 125,80: DRAW 1
25,-X: NEXT X
9030 OVER 0: POKE 23624,225: PRI
NT #0;"WARTEN SIE BITTE WEIL DAT
A IST  ""POKED""."
9050 FOR N=15616 TO 16384: POKE
(N+48984),(PEEK N): NEXT N
9055 RESTORE 9600: FOR N=(64344+
(97*8)) TO 64343+(123*8): READ A
: POKE N,A: NEXT N
9065 RESTORE 9500: FOR N=USR "A"
 TO USR "Q": READ A: POKE N,A: N
EXT N
9070 FOR Y=0 TO 14: RESTORE (975
2+2*Y): FOR N=(64000+24*Y) TO (6
4008+24*Y): READ A: POKE N,A: NE
XT N
9075 RESTORE 9799: FOR N=(64009+
24*Y) TO (64023+24*Y): READ A: P
OKE N,A: NEXT N
9080 NEXT Y
9085 POKE 23624,95: INPUT #0: PR
INT #0;"DrOcken Sie < ENTER > ei
n fOr   eine Forsetzung des Prog
rams."
9090 RANDOMIZE USR 64096: IF INK
EY$="" THEN RANDOMIZE USR 64144:
 IF INKEY$="" THEN RANDOMIZE USR
 64192: IF INKEY$="" THEN RANDOM
IZE USR 64024: IF INKEY$="" THEN
 GO TO 9090
9095 POKE 23624,56
9100 INK 1: PAPER 5: BRIGHT 0: B
ORDER 5: CLS
9105 PRINT BRIGHT 1; PAPER 4; IN
K 2;AT 2,12;"▀▀▀▀▀▀▀▀▀▀";AT 0,
12;"▄▄▄▄▄▄▄▄▄▄": PRINT PAPER 5
; INK 2;AT 1,12;"▌ENTWEICHEN▐"
9110 PRINT INK 1;AT 1,7;"F";AT 2
,7;"G";AT 1,28;"F";AT 2,28;"G"
9120 PRINT : PRINT : LET a$="Sie
 sind falsch verhaftet, und  des
wegen mNchten Sie entweichen.":
GO SUB 9800
9125 PRINT : PRINT : LET a$="Um
entweichen zu kNnnen sollen  Sie
 versuchen vier verschiedene ""A
uftrMge"" auszufOhren.": GO SUB
9800
9130 PRINT : PRINT : LET a$="Die
se vier AuftrMge sind geteiltObe
r vier Schirme.": GO SUB 9800: P
RINT : LET a$="Sie mOssen also d
urch vier ver- schiedene Schirme
 gehen, damit  Sie entweichen kN
nnen.": GO SUB 9800
9135 PRINT : PRINT : LET a$="Sie
 bekommen nur drei Leben alsosei
en Sie Vorsicht ! ! !": GO SUB 9
800: GO SUB 9810
9140 PAPER 4: BORDER 4: CLS : PR
INT AT 0,10; PAPER 7;"SCHIRM EIN
S:": PRINT INK 6;AT 1,10;"▀▀▀▀▀▀
▀▀▀▀"
9145 PRINT : LET a$="Sie sind im
 Hof des GefMngnis,  und Sie sol
len die understen    Steine des
Mauers Oberbringen    (vom Mauer
zum Mauer).": GO SUB 9800
9150 PRINT : PRINT : LET a$="Abe
r im Mitten des Hofes gibt's ein
 Graben Ober den Sie springensol
len.": GO SUB 9800: PRINT : LET
a$="Wenn Sie nicht springen, ert
rin-ken Sie und verlieren Sie ei
n   Leben.": GO SUB 9800
9155 PRINT : PRINT : LET a$="Neb
en dem Graben gibt es zwei   Sch
einwefer. Wenn Sie sich im   Leu
cht dieser Scheinwerfer      ste
hen, verlieren Sie wieder einLeb
en.": GO SUB 9800: GO SUB 9810
9160 PAPER 5: BORDER 5: CLS : IN
K 1: PRINT AT 0,10;"SCHIRM ZWEI":
 PRINT AT 1,10;"▀▀▀▀▀▀▀▀▀▀▀"
9165 LET a$="Sie sind in einem I
rrgarten an- gekommen. In diesem
 Irrgarten     gibt es POnkte, sie
 sollen alle POnkte aufessen, je
 schneller     Sie dies machen, je
 grNPer der    Bonus.": GO SUB 980
0
9170 PRINT : PRINT : LET a$="Wen
n alle POnkte verschwunden    sin
d sollen Sie zuerst zum Ein- gan
g des Irrgartens gehen und    ers
t dann sobald wie mNglich zumAus
gang. Damit Sie aus dem Irr- gar
ten entweichen kNnnen.": GO SUB
9800
9175 PRINT : PRINT : LET a$="Ach
tung ! Wenn alle POnkte       ver
schwunden sind, werden die    Kor
ridoren abgeschlossen. Sie    mOs
sen sich beeilen zum Ausgang,son
st ist er versperrt.": GO SUB 98
00: GO SUB 9810
9180 PAPER 0: BORDER 0: CLS : BR
IGHT 1: INK 7: PRINT AT 0,10;"SC
HIRM DREI";AT 1,10;"▀▀▀▀▀▀▀▀▀▀▀"
9185 PRINT : PRINT : LET a$="Bei
 diesem Schirm gibt es eine  BrO
cke, die Sie aufbauen sollen.":
GO SUB 9800
9187 PRINT : LET a$="WMhrend des
 Aufbaues kNnnen Sie erspritzt w
```

Basic-Listing »Entweichen« (Fortsetzung)

```
en.": GO SUB 9800
9190 PRINT : PRINT : LET a$="Wen
n die BrOcke fertig ist      sol
len Sie, wenn ein Leuchtblitzkom
mt, von der BrOcke ins rote  Boo
t springen.": GO SUB 9800
9195 PRINT : LET a$="Wenn Sie ni
cht richtig springen verlieren S
ie wieder ein Leben.": GO SUB 98
00: GO SUB 9810
9205 PAPER 6: INK 1: BRIGHT 0: B
ORDER 6: CLS : PRINT AT 0,10;"SC
HIRM VIER";AT 1,10;"           "
9210 PRINT : PRINT : LET a$="Auf
 diesem Schirm gibt es eine  Tre
ppe. Wenn Sie oben angekommensin
d, sind Sie entweicht !": GO SUB
 9800
9215 PRINT : LET a$="Aber, von d
ieser Treppe rollen  Ballen. Wen
n Sie von einem      dieser BMll
e getroffen werden   verlieren S
ie ein Leben. Sie    kNnnen dies
 verhOten durch in   die HNhe zu
 springen.": GO SUB 9800
9220 PRINT : PRINT : LET a$="Dan
n und wann kNnnen Sie gepacktwer
den von einem Robot (einen   grO
nen Leuchblitz kommt dann). Sie
 kNnnen dies verhindern durchin
die HNhe zu springen.": GO SUB 9
800: GO SUB 9810
9225 BRIGHT 0: PAPER 2: BORDER 2
: CLS : INK 7: PRINT AT 0,5;"FUN
KTIONEN DER TASTEN";AT 1,5;"
                     "
9230 PRINT '"SCHIRM 1: hinunter:
 6 ";TAB 10;"herauf:    7";TAB 10
;"springen: ""SPACE"""
9235 PRINT "SCHIRM 2: hinunter:
6";TAB 10;"herauf:    7";TAB 10;"
nach links:  5";TAB 10;"nach rec
hts: 8"
9240 PRINT "SCHIRM 3: hinunter:
6";TAB 10;"herauf:    7";TAB 10;"
nach links:  5";TAB 10;"nach rec
hts: 8";TAB 10;"Stein aufziehen:
 9";TAB 10;"springen: 4"
9245 PRINT "SCHIRM 4: hinunter:
6";TAB 10;"herauf:    7";TAB 10;"
springen: ""SPACE"""
9250 POKE 23624,31: PRINT ;#0;"D
rOcken Sie <ENTER> ein fOr den A
nfang des Programms.": PAUSE 0:
RANDOMIZE USR 64120: POKE 23606,
88: POKE 23607,251: GO TO 1000
9500 REM DATA FUER UDG
9505 DATA 255,0,0,0,0,0,0,0: REM
 A
9507 DATA 255,16,16,16,16,16,16,
16: REM B
9509 DATA 255,1,2,4,4,8,8,16: RE
M C
9511 DATA 0,0,0,31,31,31,31,31:
REM D
9513 DATA 31,31,31,31,0,0,0,0: R
EM E
9515 DATA 0,14,30,126,4,30,31,63
: REM F
9517 DATA 255,207,143,27,27,27,2
43,103: REM G
9518 DATA 0,3,3,3,3,4,0,0: REM H
9520 DATA 1,3,7,254,254,254,254,
254: REM I
9522 DATA 0,192,192,224,240,124,
126,254: REM J
9524 DATA 223,155,27,27,27,27,63
,255: REM K
9526 DATA 1,2,7,31,31,63,63,127:
 REM L
9527 REM DATA FUER BUCHSTABEN
9528 DATA 68,0,56,4,60,68,60,0:
REM M
9530 DATA 68,0,56,68,68,68,56,0:
 REM N
9532 DATA 68,0,68,68,68,68,56,0:
 REM O
9534 DATA 0,60,66,78,66,66,76,64
9600 REM DATA FUER 2. CHAR SET
9605 DATA 192,224,240,240,240,25
2,254,255: REM a
9607 DATA 7,15,31,63,127,255,255
,255: REM b
9609 DATA 224,240,248,252,254,25
5,255,255: REM c
9611 DATA 1,3,7,15,31,63,127,255
: REM d
9613 DATA 128,192,224,240,248,25
2,254,255: REM e
9615 DATA 0,0,0,0,60,127,255,255
: REM f
9617 DATA 255,254,252,248,240,22
4,192,128: REM g
9619 DATA 255,127,63,31,15,7,3,1
: REM h
9621 DATA 56,66,16,124,122,186,4
0,44: REM e
9623 DATA 255,255,253,249,241,22
5,193,129: REM j
9625 DATA 56,60,56,25,18,58,124,
255: REM k
9627 DATA 124,124,124,108,204,20
4,204,238: REM l
9629 DATA 192,240,56,31,31,56,24
0,192: REM m
9631 DATA 24,24,24,60,126,162,19
5,195: REM n
9633 DATA 28,124,28,24,136,188,1
24,60: REM o
9635 DATA 60,124,108,108,108,108
,236,220: REM p
9637 DATA 3,3,6,6,6,3,1,1: REM q
9639 DATA 3,7,3,3,3,3,3,2: REM r
9641 DATA 224,240,48,48,48,48,56
,172: REM s
9643 DATA 255,129,129,129,129,12
9,129,129: REM t
9645 DATA 129,129,129,129,129,12
9,129,255
9647 DATA 36,34,34,68,68,34,34,1
7: REM v
9649 DATA 60,126,255,255,255,255
,126,60: REM w
9651 DATA 255,56,56,56,56,124,25
4,255: REM x
9653 DATA 16,16,16,16,146,84,56,
16: REM y
9655 DATA 255,129,129,129,129,12
9,129,255: REM z
9750 REM DATA FUER MUSIC
9752 DATA 1,20,23,33,100,3,17,10
,0
9754 DATA 1,200,10,33,5,0,17,10,
0
9756 DATA 1,100,95,33,100,0,17,5
0,0
9758 DATA 1,40,30,33,25,1,17,50,
0
9760 DATA 1,7,30,33,37,3,17,5,0
9762 DATA 1,10,30,33,24,1,17,1,0
9764 DATA 1,10,55,33,24,2,17,1,0
9766 DATA 1,20,25,33,10,0,17,5,0
9768 DATA 1,1,205,33,10,0,17,5,0
9770 DATA 1,2,111,33,0,1,17,1,0
9772 DATA 1,200,100,33,90,1,17,0
,0
9798 REM DATA FUER MUSIC
         PROGRAM
9799 DATA 229,213,197,205,181,3,
193,209,225,125,145,111,16,242,2
01
9800 FOR N=1 TO LEN A$: PRINT A$
(N);: NEXT N: RETURN
9810 POKE 23624,31: PRINT ;#0;"D
rOcken Sie <ENTER> ein fOr eineF
orsetzung des Programms.": PAUSE
 0: RANDOMIZE USR 64120: RETURN
```

Basic-Listing »Entweichen« (Schluß)

Defender

Da fliegt man nun friedlich gesinnt durchs Weltall und wird doch tatsächlich angegriffen. Pazifisten können ausweichen, Sie dürfen sich wehren.

Wenn Sie mein Programm geladen haben, erscheinen auf dem Bildschirm am oberen Rand 12 feindliche Aliens die sich von rechts nach links bewegen. Plötzlich bricht ein feindliches Raumschiff aus und versucht Sie mit Bomben zu treffen. In dieser Spielsituation haben Sie nur die Möglichkeit ihre Raumbasis mit den Tasten 5 nach links und der Taste 8 nach rechts zu lenken.

Sie können ihm also nur ausweichen, ihn aber nicht vernichten. Erst wenn der Angreifer verschwunden ist können Sie mit Hilfe der Taste 7 auf die Invasoren schießen (immer nur ein Schuß). Da die Angreifer immer näher auf Sie zukommen, wird das Spiel mit der Zeit immer schwieriger. Schaffen Sie es alle Aliens mit ihren 3 Raumbasen abzuschießen erscheinen neue Invasoren, die aber jetzt schon sehr viel tiefer fliegen.

hi	=	High-Score Punktzahl
sum	=	Punktzahl im Spiel
live	=	Anzahl der vorhandenen Raumbasen
lo	=	Variable für die Anfangshöhe in der die Aliens fliegen.
q	=	Variable der Höhe der oberen Reihe der Invasoren
r	=	Variable der Höhe der unteren Reihe der Invasoren
x,y	=	Koordinaten der Raumbasis
o,y	=	Koordinaten der Rakete der Raumbasis
w	=	y-Koordinate der einzelnen Angreifer
a$ —		
x$	=	Variable der einzelnen feindlichen Raumschiffe
s,g	=	Koordinaten des angreifenden Aliens

Variabledefinition für das Programm Defender

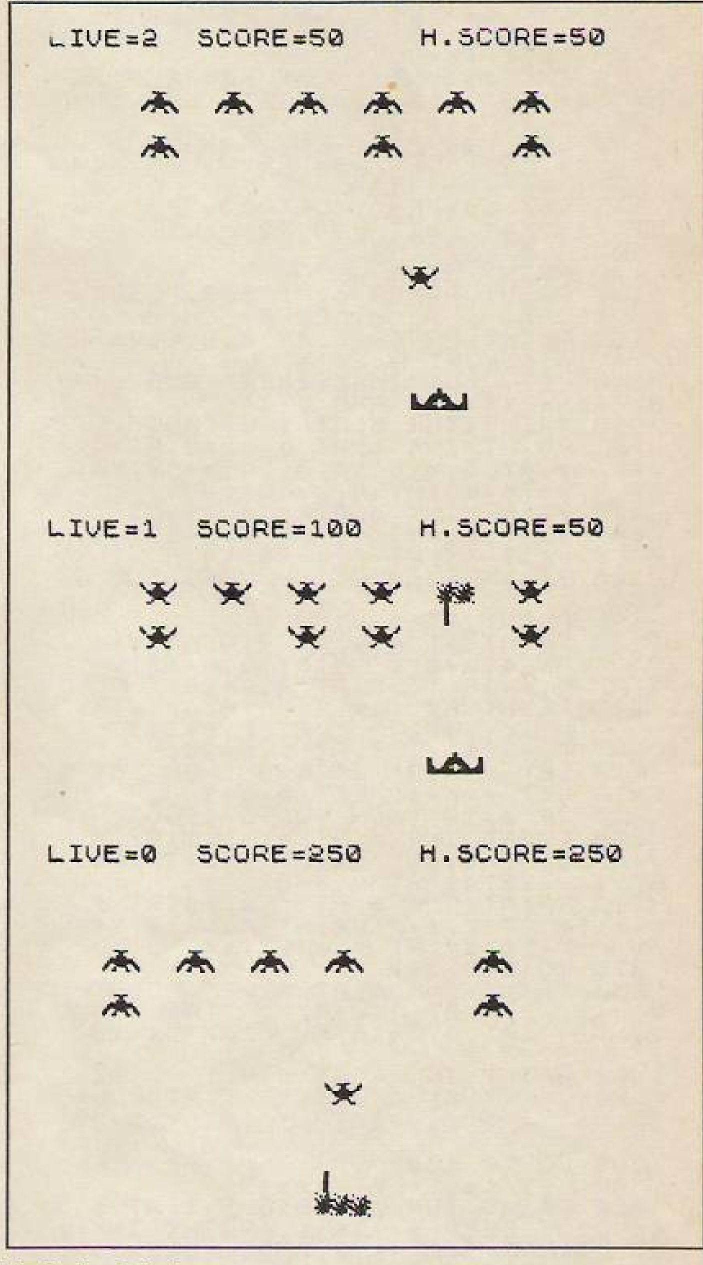
10 — 41	USER-Grafik wird festgelegt
45 — 70	Variabledefinition
140	Tastaturabfrage der Tasten 5 und 8 (links — rechts)
160 — 290	Darstellung der Raumbasis und der 12 Invasoren auf dem Bildschirm.
195	Tastaturabfrage der Taste 7 (Feuertaste)
300 — 360	Bewegung des angreifenden Aliens, der aus der Staffel feindlicher Raumschiffe ausbricht
510 — 570	Bombenabwurf auf die Raumbasis und nochmalige Tastaturabfrage der Tasten 5 und 8
1500 — 1510	Abzug einer Raumbasis
2000 — 2050	Spielende mit Anzeige des High-Score und der Frage nach einem weiterem Spiel.
5000 — 5180	Bewegung der abgefeuerten Rakete der Raumbasis. Mit Überprüfung, ob ein Treffer erzielt wurde.

Die einzelnen Programmabschnittes des Spiels Defender

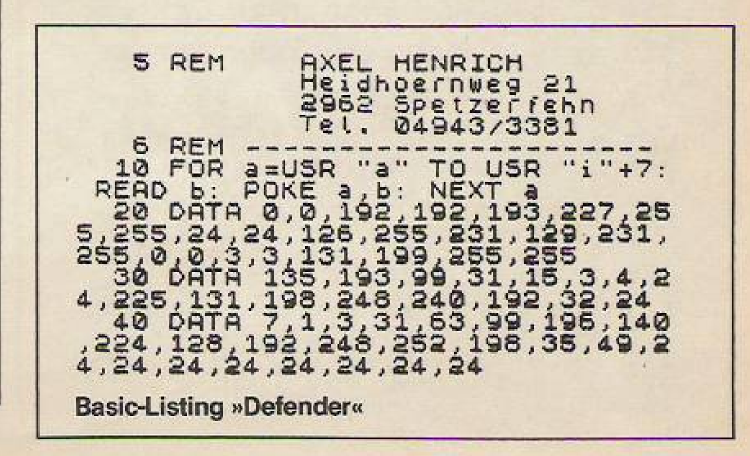
Tastenfunktionen beim Spiel Defender
"5" nach links
"8" nach rechts
"7" Feuer

Das Programm »Defender« kann mit »SAVE"DEFENDER" LINE 1« aufgenommen und von Kassette mit LOAD geladen werden.

(A. Henrich)



Typische Spielszenen



```
   5 REM    AXEL HENRICH
            Heidhoernweg 21
            2962 Spetzerfehn
            Tel. 04943/3381
   6 REM -----------------------
  10 FOR a=USR "a" TO USR "i"+7:
 READ b: POKE a,b: NEXT a
  20 DATA 0,0,192,192,193,227,25
5,255,24,24,126,255,231,129,231,
255,0,0,3,3,131,199,255,255
  30 DATA 135,193,99,31,15,3,4,2
4,225,131,198,248,240,192,32,24
  40 DATA 7,1,3,31,63,99,196,140
,224,128,192,248,252,198,35,49,2
4,24,24,24,24,24,24,24
```

Basic-Listing »Defender«

```
  41 DATA 132,44,151,92,59,220,4
4,81
  45 LET hi=0
  50 BORDER 1: PAPER 1: INK 7: C
LS
  52 LET lo=0: LET q=1: LET r=3:
 LET live=3: LET sum=0
  55 PRINT PAPER 4;AT 0,0;"LIVE=
   SCORE=        H.SCORE=      "
  60 LET o=0: LET w=1: LET x=21:
 LET y=10
  70 LET a$="DE": LET b$="DE": L
ET c$="DE": LET d$="DE": LET e$=
"DE": LET f$="DE"
  75 LET g$="DE": LET h$="DE": L
ET i$="DE": LET j$="DE": LET k$=
"DE": LET l$="DE"
  80 LET m$="FG": LET n$="FG": L
ET o$="FG": LET p$="FG": LET q$=
"FG": LET r$="FG"
  85 LET s$="FG": LET t$="FG": L
ET u$="FG": LET v$="FG": LET w$=
"FG": LET x$="FG"
 100 PRINT AT x,y;"   "
 105 PRINT PAPER 4;AT 0,0;"LIVE=
   SCORE=        H.SCORE=      "
 110 PRINT PAPER 4;AT 0,5;live;A
T 0,14;sum;AT 0,28;hi
 140 LET y=y+(INKEY$="8" AND y<2
8)-(INKEY$="5" AND y>1)
 160 PRINT INK 5;AT x,y;"ABC"
 180 PRINT INK 4;AT q,w;a$;AT q,
w+4;b$;AT q,w+8;c$;AT q,w+12;d$;
AT q,w+16;e$;AT q,w+20;f$
 190 PRINT INK 5;AT r,w;g$;AT r,
w+4;h$;AT r,w+8;i$;AT r,w+12;j$;
AT r,w+16;k$;AT r,w+20;l$
 195 IF INKEY$="7" THEN GO SUB 5
000
 196 PRINT AT o,y;" "
 200 PRINT AT q,w;"  ";AT q,w+4;
"  ";AT q,w+8;"  ";AT q,w+12;"
";AT q,w+16;"  ";AT q,w+20;"  "
 210 PRINT AT r,w;"  ";AT r,w+4;
"  ";AT r,w+8;"  ";AT r,w+12;"
";AT r,w+16;"  ";AT r,w+20;"  "
 220 LET w=w+2: IF w=9 THEN LET
w=1: LET r=r+1: LET q=q+1
 230 IF r=19 THEN GO TO 1000
 260 PRINT INK 4;AT q,w;m$;AT q,
w+4;n$;AT q,w+8;o$;AT q,w+12;p$;
AT q,w+16;q$;AT q,w+20;r$
 270 PRINT INK 5;AT r,w;s$;AT r,
w+4;t$;AT r,w+8;u$;AT r,w+12;v$;
AT r,w+16;w$;AT r,w+20;x$
 275 GO SUB 300
 280 PRINT INK 4;AT q,w;"  ";AT
q,w+4;"  ";AT q,w+8;"  ";AT q,w+
12;"  ";AT q,w+16;"  ";AT q,w+20
;"  "
 290 PRINT INK 4;AT r,w;"  ";AT
r,w+4;"  ";AT r,w+8;"  ";AT r,w+
12;"  ";AT r,w+16;"  ";AT r,w+20
;"  "
 295 GO TO 100
 300 LET s=r: LET g=w+8
 310 PRINT INK 2; BRIGHT 1;AT s,
g;"DE": BEEP 1/100,s: PRINT INK
2; BRIGHT 1;AT s,g;"FG": BEEP 1/
100,g: PRINT AT s,g;"  "
 320 LET s=s+1
 330 IF g>y AND s<18 THEN LET g=
g-1
 340 IF g<y AND s<18 THEN LET g=
g+1
 345 IF lo=12 THEN LET live=live
+1: LET lo=13: LET q=3: LET r=5:
 LET sum=sum+500: CLS : GO TO 55
 346 IF lo=25 THEN LET live=live
+1: LET lo=26: LET q=5: LET r=7:
 LET sum=sum+500: CLS : GO TO 55
 350 IF g=y THEN PRINT INK 2; BR
IGHT 1;AT s,g;"DE": GO SUB 500
 360 IF s=21 THEN RETURN
 370 GO TO 310
 510 FOR a=s+1 TO 20
 515 PRINT INK 5;AT x,y;"   "
 520 PRINT INK 3;AT a,g;"H"
 530 IF a+1=x AND g=y THEN GO TO
 1500
 535 LET y=y+(INKEY$="8" AND y<2
8)-(INKEY$="5" AND y>1)
 536 PRINT INK 5;AT x,y;"ABC"
 550 PRINT AT a,g;" "
 560 NEXT a
 565 PRINT AT s,g;"  "
 570 RETURN
1000 PRINT AT 10,2;"Die Invasion
 ist gelungen !"
1500 LET live=live-1: BEEP 1/100
,40
1510 IF live<=0 THEN GO TO 2000
1520 GO TO 536
2000 IF sum>hi THEN LET hi=sum
2010 PRINT AT x,y;"III"
2020 PRINT PAPER 4;AT 0,5;live;A
T 0,14;sum;AT 0,28;hi
2030 INPUT "NEW GAME (y/n)?";z$
2040 IF z$="y" THEN GO TO 50
2050 GO TO 9000
5010 LET o=x-1
5020 PRINT INK 6;AT o,y;"H": BEE
P 1/1000,o*3
5030 IF o-1=q AND (y=w OR y=w+1)
 THEN IF a$="DE" THEN PRINT AT q
,w;"II": BEEP 1/10,-10: LET lo=l
o+1: LET sum=sum+100: LET a$="
": LET m$="  ": RETURN
5040 IF o-1=q AND (y=w+4 OR y=w+
5) THEN IF b$="DE" THEN PRINT AT
 q,w+4;"II": BEEP 1/10,-10: LET
lo=lo+1: LET sum=sum+100: LET b$
="  ": LET n$="  ": RETURN
5050 IF o-1=q AND (y=w+8 OR y=w+
9) THEN IF c$="DE" THEN PRINT AT
 q,w+8;"II": BEEP 1/10,-10: LET
lo=lo+1: LET sum=sum+100: LET c$
="  ": LET o$="  ": RETURN
5060 IF o-1=q AND (y=w+12 OR y=w
+13) THEN IF d$="DE" THEN PRINT
AT q,w+12;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+100: LET
 d$="  ": LET p$="  ": RETURN
5070 IF o-1=q AND (y=w+16 OR y=w
+17) THEN IF e$="DE" THEN PRINT
AT q,w+16;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+100: LET
 e$="  ": LET q$="  ": RETURN
5080 IF o-1=q AND (y=w+20 OR y=w
+21) THEN IF f$="DE" THEN PRINT
AT q,w+20;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+100: LET
 f$="  ": LET r$="  ": RETURN
5090 IF o-1=r AND (y=w OR y=w+1)
 THEN IF g$="DE" THEN PRINT AT r
,w;"II": BEEP 1/10,-10: LET lo=l
o+1: LET sum=sum+50: LET g$="  "
: LET s$="  ": RETURN
5100 IF o-1=r AND (y=w+4 OR y=w+
5) THEN IF h$="DE" THEN PRINT AT
 r,w+4;"II": BEEP 1/10,-10: LET
lo=lo+1: LET sum=sum+50: LET h$=
"  ": LET t$="  ": RETURN
5110 IF o-1=r AND (y=w+8 OR y=w+
9) THEN IF i$="DE" THEN PRINT AT
 r,w+8;"II": BEEP 1/10,-10: LET
lo=lo+1: LET sum=sum+50: LET i$=
"  ": LET u$="  ": RETURN
5120 IF o-1=r AND (y=w+12 OR y=w
+13) THEN IF j$="DE" THEN PRINT
AT r,w+12;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+50: LET
j$="  ": LET v$="  ": RETURN
5130 IF o-1=r AND (y=w+16 OR y=w
+17) THEN IF k$="DE" THEN PRINT
AT r,w+16;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+50: LET
k$="  ": LET w$="  ": RETURN
5140 IF o-1=r AND (y=w+20 OR y=w
+21) THEN IF l$="DE" THEN PRINT
AT r,w+20;"II": BEEP 1/10,-10: L
ET lo=lo+1: LET sum=sum+50: LET
l$="  ": LET x$="  ": RETURN
5160 IF o<q THEN RETURN
5170 PRINT AT o,y;" ": LET o=o-1
5180 GO TO 5020
9000 STOP
```

Basic-Listing »Defender« (Schluß)

Anaconda ...
... oder Fütterung der Raubtiere am Beispiel einer Schlange

Ein Schlangenspiel mit konsequenter Nutzung der UDGs, die im Listing als Buchstabe mit einem Strich darunter ausgedruckt wurden.

Nachdem das Programm geladen wurde, erscheint das Titelbild. Sobald man eine Taste gedrückt hat, kann man die Tasten wählen, mit denen man spielen möchte. Danach beginnt das Spiel. Es handelt davon, eine immer länger werdende Schlange durch ein Labyrinth kriechen und blinkende Punkte auffressen zu lassen. Nach einer gewissen Zeit wird ein Bonus zu den Punkten addiert und eine neue Spielstufe beginnt. Sobald sich die Schlange nicht mehr bewegen kann oder alle 5 vorhandenen Labyrinthe geschafft sind, ist das Spiel zu Ende und man wird zu einem neuen Spiel aufgefordert. (S. Gleissner)

```
   1 REM ANACONDA
   2 REM © by SIMON GLEISSNER
   3 REM 17.8.1984
   4 PRINT AT 10,10; FLASH 1;"Bi
tte warten": GO TO 45
   9 LET hi=0: LET tt=200: LET q
2=10: LET q6=10
  10 DIM o(1000): DIM p(1000)
  20 BORDER 0: PAPER 0: INK 7: C
LS : PRINT AT 0,0;"FREIE TASTENW
AHL FUER RICHTUNGS-AENDERUNG DER
 SCHLANGE"
  21 INPUT "Oben ";e$: IF LEN e$
<>1 THEN GO TO 21
  22 INPUT "Unten ";b$: IF LEN b
$<>1 THEN GO TO 22
  23 INPUT "Rechts ";c$: IF LEN
c$<>1 THEN GO TO 23
  24 INPUT "Links ";d$: IF LEN d
$<>1 THEN GO TO 24
  29 CLS : PRINT AT 0,0;"ANACOND
A";AT 4,0;"<";e$;">  Oben";AT 6,
0;"<";b$;">  Unten";AT 8,0;"<";d
$;">  Links";AT 10,0;"<";c$;">
Rechts": FOR w=1 TO 100: NEXT w:
 PRINT AT 15,0;"Bitte eine Taste
 druecken": PAUSE 0
  30 GO TO 551
  45 REM ██ AUFBAU DER SCHLANGE
  50 FOR t=0 TO 7
  60 READ n
  70 POKE USR "a"+t,n
  80 NEXT t
  90 DATA 60,126,219,255,255,219
,126,60
 100 FOR t=0 TO 7
 105 READ n
 110 POKE USR "b"+t,n
 120 NEXT t
 130 DATA 60,126,219,126,60,24,0
,0
 140 FOR t=0 TO 7
 150 READ n
 160 POKE USR "c"+t,n
 170 NEXT t
 180 DATA 0,0,24,60,126,219,126,
60
 190 FOR t=0 TO 7
 200 READ n
 210 POKE USR "d"+t,n
 220 NEXT t
 230 DATA 32,112,216,252,252,216
,112,32
 240 FOR t=0 TO 7
 250 READ n
 260 POKE USR "e"+t,n
 270 NEXT t
 280 DATA 4,14,23,63,63,23,14,4
 290 FOR t=0 TO 7
 300 READ n
 310 POKE USR "f"+t,n
 320 NEXT t
 330 DATA 24,24,24,24,60,60,60,6
0
 340 FOR t=0 TO 7
 350 READ n
 360 POKE USR "g"+t,n
 370 NEXT t
 380 DATA 60,60,60,60,24,24,24,2
4
 390 FOR t=0 TO 7
 400 READ n
 410 POKE USR "h"+t,n
 420 NEXT t
 430 DATA 0,0,240,255,255,240,0,
0
 440 FOR t=0 TO 7
 450 READ n
 460 POKE USR "i"+t,n
 470 NEXT t
 480 DATA 0,0,15,255,255,15,0,0
 500 REM ██ IRRGARTEN
 510 FOR t=0 TO 7
 520 READ n
 530 POKE USR "j"+t,n
 540 NEXT t
 541 GO TO 3000
 550 DATA 255,153,153,255,255,15
3,153,255
 551 LET pu=0
 600 IF q2=0 THEN FOR w=1 TO 300
: NEXT w: PRINT PAPER 6; INK 0;A
T 0,0;"Gesamte Punktezahl:
     ";AT 0,19;pu: GO TO 4100
 601 IF q6=10 THEN BORDER 0: PAP
ER 0: INK 7: CLS
 602 IF q6=8 THEN BORDER 6: PAPE
R 6: INK 1: CLS
 603 IF q6=6 THEN BORDER 2: PAPE
R 2: INK 5: CLS
 604 IF q6=4 THEN BORDER 7: PAPE
R 7: INK 3: CLS
 605 IF q6=2 THEN BORDER 6: PAPE
R 6: INK 2: CLS
 608 PRINT AT 0,0: IF q6<8 THEN
GO TO 811
 609 REM J='j' im Graphic-Mode
 610 PRINT AT 1,0;"JJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJ"
 620 PRINT AT 2,0;"J          J
  JJ         J          J"
 630 PRINT AT 3,0;"J JJJJ J JJJJ
J JJ JJJJJ J JJJJ J"
 640 PRINT AT 4,0;"J J
                       J   J"
 650 PRINT AT 5,0;"J J JJ J JJJ
J JJ J JJJ J JJ J J"
 660 PRINT AT 6,0;"J J JJ J JJJ
J JJ J JJJ J JJ J J"
 670 PRINT AT 7,0;"J J JJ J
J JJ J        J JJ J J"
 680 PRINT AT 8,0;"J J JJ JJJJJ
J JJ J JJJJJ JJ J J"
 690 PRINT AT 9,0;"J
  JJ                    J"
```

Basic-Listing »Anaconda«

```
 700 PRINT AT 10,0;"J J JJJJJJJJ
JJJJJJ JJJJJJJJJ J J"
 710 PRINT AT 11,0;"J
 720 PRINT AT 12,0;"J JJJJJJJJ J
JJJJJJ J JJJJJJJJ J"
 730 PRINT AT 13,0;"J                J
 740 PRINT AT 14,0;"J JJJJJJJJ J
JJJJJJJJJJ JJJJJJJJJ J"
 750 PRINT AT 15,0;"J
 760 PRINT AT 16,0;"J J J J   J J
JJ JJ JJJ J  J J J J"
 770 PRINT AT 17,0;"J J J J   J J
 780 PRINT AT 18,0;"J J J J   J J
 790 PRINT AT 19,0;"J J J JJJJ J
 800 PRINT AT 20,0;"J   J        J
 810 PRINT AT 21,0;"JJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJ": GO TO 849
 811 IF q6=2 THEN GO TO 815
 813 PRINT AT 1,0;"JJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJ
 JJJJ JJJ J J J J J J J J J JJJ
 JJJJ J JJJ J J J J J J J JJJ J
 JJJJ J JJJ J J J J J J J JJJ J
 JJJJ JJJJJ J JJJ J JJJ J JJJJJ
JJJJJJ JJJJ J J J J J J J JJJJ
JJJJJJ J JJ J J J J J J J JJ J
JJJJJJ J JJJJ J J J J J JJJJ J
J JJ JJJJ J JJJ J J J JJJ J JJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ"
 814 GO TO 849
 815 PRINT AT 1,0;"JJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJ": FOR w=2 TO
 18 STEP 2: PRINT AT w,0;"J ";AT w
+1,0;"J J J J J J J J J J J J J
J J J ": NEXT w: PRINT AT 20,0;"
": PRINT AT 21,0;"JJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJ"
 820 PRINT AT 10,17; INK 4;CHR$ 
152;CHR$ 147: GO TO 850
 849 PRINT AT 11,17; INK 4;CHR$ 
152;CHR$ 147
 850 LET u1=2+INT (RND*19): LET 
u2=1+INT (RND*30): IF SCREEN$ (u
1,u2)<>" " THEN GO TO 850
 860 PRINT AT u1,u2; INK 4; PAPE
R 2; FLASH 1;" "
 870 LET o(1)=11: LET o(2)=11: L
ET p(1)=17: LET p(2)=18: IF q6=2
 THEN LET o(1)=10: LET o(2)=10
 871 LET q1=1: LET q3=3
 872 LET q4=2: LET q5=1
 880 RESTORE 900
 881 READ vv1,vv2: IF vv1=200 TH
EN GO TO 901
 882 BEEP vv1,vv2: GO TO 881
 900 DATA .25,-5,.5,0,.25,0,.25,
2,.5,4,.5,0,1,7,.5,4,.5,4,.5,5,.
25,7,.25,5,.25,4,.25,5,.5,7,.25,
2,.25,0,.25,2,.25,4,.5,2,.5,-5,.
5,0,.25,0,.25,2,.5,4,.5,0,1,7,.5
,4,.5,4,.25,5,.25,7,.25,4,.25,5,
.75,2,.25,0,1.25,0,200,0
 999 PRINT AT 0,0;"

1000 REM ■■ SPIEL
1001 LET x=18: LET y=11: IF q6=2
 THEN LET y=10
1002 LET a=0: LET b=1
1005 PRINT AT 0,0;"Punkte:", FLA
SH 1;hi: GO TO 2000
1010 PRINT AT 0,8;pu: LET a$=INK
EY$: IF a$=d$ THEN LET b=-1
1020 IF a$=d$ OR a$=c$ THEN LET 
a=0
1030 IF a$=c$ THEN LET b=1
1040 IF a$=b$ THEN LET a=1
1050 IF a$=e$ THEN LET a=-1
1060 IF a$=e$ OR a$=b$ THEN LET 
b=0
1063 IF y<>u1 OR x<>u2 THEN GO T
O 1070
1064 BEEP .05,16: BEEP .05,17: L
ET pu=pu+50
1065 LET u1=2+INT (RND*19): LET 
u2=1+INT (RND*30): IF SCREEN$ (u
1,u2)=" " THEN GO TO 1067
1066 GO TO 1065
1067 PRINT AT u1,u2; INK 2; PAPE
R 4; FLASH 1;" "
2000 IF SCREEN$ (y+a,x+b)=" " TH
EN GO TO 2002
2001 GO TO 1010
2010 LET y=y+a: LET x=x+b
2020 IF b=1 THEN PRINT AT y,x; I
NK 4;CHR$ 147
2030 IF b=-1 THEN PRINT AT y,x;
INK 4;CHR$ 148
2040 IF a=1 THEN PRINT AT y,x; I
NK 4;CHR$ 145
2050 IF a=-1 THEN PRINT AT y,x;
INK 4;CHR$ 146
2060 PRINT AT y-a,x-b; INK 4;CHR
$ 144: BEEP .01,0
2100 IF SCREEN$ (y+1,x)=" " THEN
 GO TO 2200
2110 IF SCREEN$ (y,x+1)=" " THEN
 GO TO 2200
2120 IF SCREEN$ (y-1,x)=" " THEN
 GO TO 2200
2130 IF SCREEN$ (y,x-1)=" " THEN
 GO TO 2200
2131 LET q=0
2140 FOR t=12 TO 36 STEP 2: BORD
ER INT (RND*8): BEEP 0.001,t: NE
XT t: LET q=q+1
2150 FOR t=36 TO 12 STEP -2: BO
RDER INT (RND*8): BEEP 0.001,t:
NEXT t: IF q<=3 THEN GO TO 2140
2160 BORDER 0: GO TO 4100
2300 LET o(q3)=y: LET p(q3)=x: I
F q3=tt THEN LET tt=tt+197: LET 
q6=q6-2: LET q2=q6: DIM o(1000):
 DIM p(1000): PRINT AT 0,0; FLAS
H 1;"
     ";AT 0,0; FLASH 1;"P.:";pu+
INT (pu/100*20);"  Bonus(20%):";
INT (pu/100*20): LET pu=pu+INT (
pu/100*20): FOR u=-20 TO 40: BEE
P .01,u: NEXT u: GO TO 600
2310 LET q3=q3+1
2320 IF q2=0 THEN LET q2=q6: GO
TO 2500
2330 LET q2=q2-1
2331 PRINT AT q4,q5;" "
2340 IF p(q1)<p(q1+1) THEN PRINT
 INK 4;AT o(q1),p(q1);CHR$ 152
2350 IF p(q1)>p(q1+1) THEN PRINT
 INK 4;AT o(q1),p(q1);CHR$ 151
2360 IF o(q1)>o(q1+1) THEN PRINT
 INK 4;AT o(q1),p(q1);CHR$ 150
2370 IF o(q1)<o(q1+1) THEN PRINT
 INK 4;AT o(q1),p(q1);CHR$ 149
2380 LET q4=o(q1): LET q5=p(q1)
2400 LET q1=q1+1
2500 LET pu=pu+1
2600 GO TO 1010
3000 REM ■■ Schriftzug
3010 BORDER 0: PAPER 0: INK 7: C
LS
3015 FOR s=-2 TO 4
3020 PLOT 10+s,130+s: DRAW 0,30:
```

Basic-Listing »Anaconda« (Fortsetzung)

```
 DRAW 10,10: DRAW 10,-10: DRAW 0
,-30: DRAW 0,20: DRAW -20,0
3030 PLOT 40+s,130+s: DRAW 0,40:
 DRAW 10,-10: DRAW 0,-20: DRAW 1
0,-10: DRAW 0,40
3040 PLOT 70+s,130+s: DRAW 0,30:
 DRAW 10,10: DRAW 10,-10: DRAW 0
,-30: DRAW 0,20: DRAW -20,0
3050 PLOT 120+s,140+s: DRAW -10,
-10: DRAW -10,10: DRAW 0,20: DRA
W 10,10: DRAW 10,-10
3060 PLOT 150+s,140+s: DRAW -10,
-10: DRAW -10,10: DRAW 0,20: DRA
W 10,10: DRAW 10,-10: DRAW 0,-20
3070 PLOT 160+s,130+s: DRAW 0,40
: DRAW 10,-10: DRAW 0,-20: DRAW
10,-10: DRAW 0,40
3080 PLOT 190+s,130+s: DRAW 0,40
: DRAW 10,0: DRAW 10,-10: DRAW 0
,-20: DRAW -10,-10: DRAW -10,0
3090 PLOT 220+s,130+s: DRAW 0,30
: DRAW 10,10: DRAW 10,-10: DRAW
0,-30: DRAW 0,20: DRAW -20,0
3100 NEXT s
3110 PRINT AT 15,5;"© by SIMON G
LEISSNER"
3115 PRINT AT 19,1;"Schlaengeln
Sie sich durch die   Labyrinthe
und erreichen Sie         die blin
kenden Punkte"
3120 DIM o(56): DIM p(56)
3121 RESTORE 3140: FOR s=1 TO 56
: READ o(s): READ p(s): NEXT s
3140 DATA 0,0,0,1,1,1,2,1,2,2,2,
3,2,4,1,4,0,4,0,5,0,6,0,7,0,8,0,
9,0,10,1,10,2,10,2,11,2,12,2,13,
2,14,2,15,2,16,2,17,2,18,2,19,2,
20,1,20,0,20,0,19,0,18,0,17,0,16
,0,15,0,14,0,13,0,12,0,11,0,10,1
,10,2,10,2,9,2,8,2,7,2,6,2,5,3,5
,3,4,3,3,3,2,3,1,3,0,3,-1,2,-1,1
,-1,0,-1
3150 INK 4: FOR w=1 TO 10: PRINT
 AT o(w)+14,p(w)+5;CHR$ 144: NEX
T w
3199 LET v=0
3200 FOR s=1 TO 56
3205 IF s=1 THEN PRINT AT 17,4;
INK 7;"17.8.84";AT o(56)+14,p(56
)+5; INK 4;CHR$ 152;AT o(55)+14,
p(55)+5;" ": GO TO 3220
3210 IF o(s-1)<o(s) THEN PRINT A
T o(s-1)+14,p(s-1)+5;CHR$ 149: I
F s<>2 THEN PRINT AT o(s-2)+14,p
(s-2)+5;" "
3211 IF o(s-1)>o(s) THEN PRINT A
T o(s-1)+14,p(s-1)+5;CHR$ 150: I
F s<>2 THEN PRINT AT o(s-2)+14,p
(s-2)+5;" "
3212 IF p(s-1)>p(s) THEN PRINT A
T o(s-1)+14,p(s-1)+5;CHR$ 151: I
F s<>2 THEN PRINT AT o(s-2)+14,p
(s-2)+5;" "
3213 IF p(s-1)<p(s) THEN PRINT A
T o(s-1)+14,p(s-1)+5;CHR$ 152: I
F s<>2 THEN PRINT AT o(s-2)+14,p
(s-2)+5;" "
3214 IF s=2 THEN PRINT AT o(56)+
14,p(56)+5;" "
3230 LET s1=s+10: IF s1>56 THEN
LET s1=s1-56
3235 PRINT AT o(s1)+14,p(s1)+5;C
HR$ 144
3240 LET s1=s1+1: IF s1>56 THEN
LET s1=s1-56
3250 IF s1=1 THEN PRINT AT o(1)+
14,p(1)+5;CHR$ 147: GO TO 3300
3260 IF p(s1-1)<p(s1) THEN PRINT
 AT o(s1)+14,p(s1)+5;CHR$ 147
3270 IF p(s1-1)>p(s1) THEN PRINT
 AT o(s1)+14,p(s1)+5;CHR$ 148
3280 IF o(s1-1)>o(s1) THEN PRINT
 AT o(s1)+14,p(s1)+5;CHR$ 146
3290 IF o(s1-1)<o(s1) THEN PRINT
 AT o(s1)+14,p(s1)+5;CHR$ 145
3500 LET v=v+1: IF INKEY$<>"" OR
 v>170 THEN GO TO 8
4000 BEEP .01,20: NEXT s
```

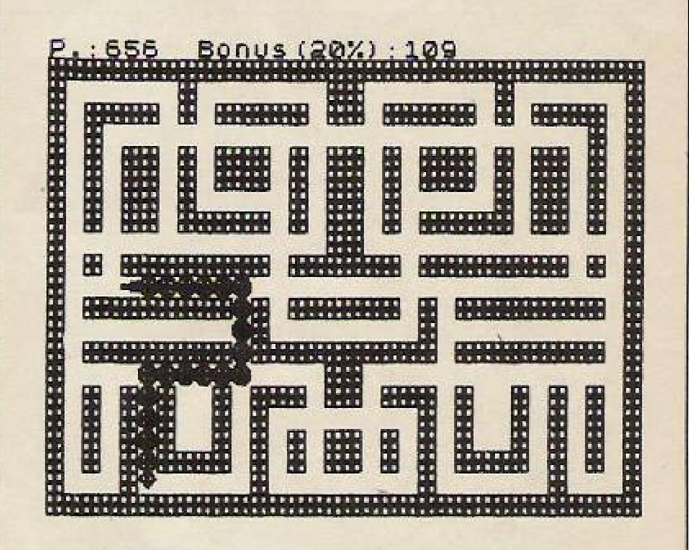
```
4001 GO TO 3200
4100 REM ██ Neues Spiel
4110 PRINT AT 21,1; FLASH 1; INK
 7; PAPER 0;"Neues Spiel (ENTER)
"
4120 IF INKEY$<>CHR$ 13 THEN GO
TO 4120
4125 IF pu>hi THEN LET hi=pu
4130 LET tt=200: LET q2=10: LET
q6=10: DIM o(1000): DIM p(1000):
 GO TO 551
```

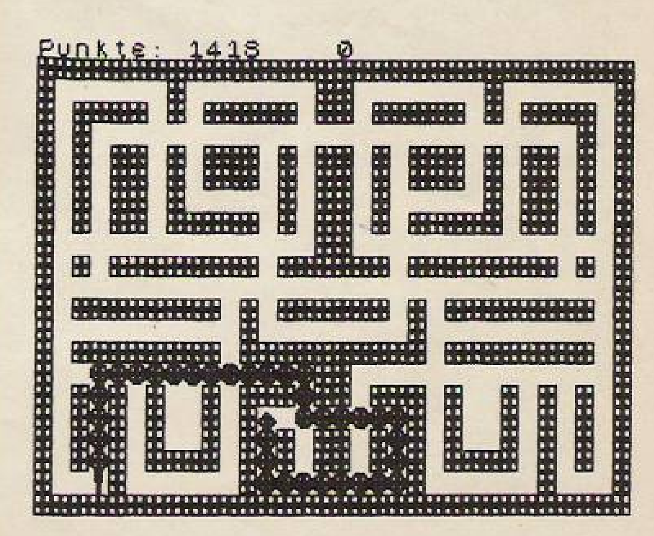
Basic-Listing »Anaconda« (Schluß)

ANACONDA

© by SIMON GLEISSNER

Schlaengeln Sie sich durch die
Labyrinthe und erreichen Sie
die blinkenden Punkte





Hardcopies »Anaconda«

Geheim

Abenteuerspiele, die in Zeitschriften abgedruckt werden, haben einen Nachteil: Der Leser erfährt beim Eintippen, wie er die Rätsel zu lösen hat. Das Programm »Geheim« für jeden Spectrum löst dieses Problem.

Mit diesem Hilfsprogramm kann man den Zeichensatz verschieben. Resultat: Soll zum Beispiel ein K auf dem Bildschirm erscheinen, muß ein I eingegeben werden.

Nachdem das Programm eingegeben und mit GO TO 300 auf Band gesichert wurde, wird es gestartet. Es erscheint der normale Zeichensatz in Schwarz und der jeweils geänderte in Rot.

Unten steht jeweils der POKE-Befehl, der eingegeben werden muß, um den »roten« Zeichensatz zu erhalten. Durch Drücken einer beliebigen Taste verschiebt sich der Zeichensatz immer mehr. Man gibt zum Beispiel folgendes Listing ein:

Durch POKE 23606,0 wird der Zeichensatz wieder »normal«. (B. Baran)

```
   95 REM © Bernhard Baran
            Postfach 35
            6710 Frankenthal
            Tel.:06233/24243
  100 REM ■ Zeichensatz mit sich
            selbst verschieben ■
  102 POKE 23606,0
  105 BORDER 4: PAPER 6: INK 3: C
LS : LET a=1: LET b=6: LET c=11:
 LET i=0
  110 PRINT AT 17,0;"SCHWARZ=norm
aler Zeichensatz";AT 19,0;"ROT=d
urch untenstehenden POKE-";AT 20
,0;"Befehl veraenderten Zeichens
atz": GO SUB 120
  115 GO TO 210
  120 LET q=0
  125 FOR g=32 TO 63
  130 PRINT AT a,q; INK i;CHR$ g;
: LET q=q+1
  140 NEXT g
  150 LET q=0
  155 FOR h=64 TO 95
  160 PRINT AT b,q; INK i;CHR$ h;
: LET q=q+1
  170 NEXT h
  180 LET q=0
  185 FOR j=96 TO 127
  190 PRINT AT c,q; INK i;CHR$ j;
: LET q=q+1
  200 NEXT j
  205 RETURN
  210 FOR f=0 TO 255
  220 POKE 23606,f
  230 LET a=3: LET b=8: LET c=13:
 LET i=2: GO SUB 120
  240 POKE 23606,0
  250 PRINT #0;"POKE 23606,";f
  260 PAUSE 0: PRINT #0;AT 0,0;"
              ": NEXT f: STOP
  300 SAVE "Geheim" LINE 1
  310 PRINT "Band zurueckspulen f
uer Verify"
  320 VERIFY "Geheim"
```

Basic-Listing »Geheim«

102	setzt Zeichensatz auf Normalstellung
105	Farbsetzung und Variablenbestimmung
110	Informationen erscheinen auf dem Bildschirm
120 — 205	kompletter Zeichensatz erscheint auf dem Bildschirm
210 — 260	POKE-Befehl-Schleife
240 — 250	neuer POKE-Befehl erscheint auf dem Bildschirm
260	Tastaturabfrage
300 — 320	SAVEn für Programm: »Geheim«

Programmstruktur

```
1000>POKE 23606,16
1010 PRINT "Bgc©Rcvrqrcjjcl©gl©?
£clrcscp+©©©qngcjcl©imcllcl©lsl©
lgafr©kcfp©©xsk©Jmcgcl©cglcq©P_c
rqcjq©kggq+©£p_safr©ucpbcl,"
1020 POKE 23606,0
```

Verschleierter Text

a	1. X-Print-Position für Zeichensatz
b	2. X-Print-Position für Zeichensatz
c	3. X-Print-Position für Zeichensatz
i	jeweilige INK-Farbe
q	Y-Print-Position für Zeichensatz
f	POKE-Befehl-Schleifenvariable
g	1. Schleife zum Zeichnen des Zeichens
h	2. Schleife zum Zeichnen des Zeichens
j	3. Schleife zum Zeichnen des Zeichens

Variablen-Tabelle

Tasword-Einzeiler

Immer mehr Spectrum-Fans kaufen sich ein Interface 1 und steuern dann mit der RS232-Schnittstelle einen Drucker an.

Tasword two, »das« Textverarbeitungsprogramm, ist jedoch für ein Centronics-Interface geschrieben. Fügt man die Zeile 265 ein, kann man jedoch auch seriell ausdrucken. Die Interface-Kontrollcodes können dann auf »0« gesetzt werden. Eventuell (darüber gibt das Druckerhandbuch Auskunft) müssen andere Werte eingestellt werden. Beim Epson FX-80 und beim GLP von Centronics reicht »0«. (mk)

```
265 CLEAR #: FORMAT "b";9600:
    OPEN #3;"b"
```

Basic-Zeile 265

Space-Smily

Hier ist ein 5-Minuten-Spielchen für alle Spectrum-User, die mit Beta-Basic arbeiten. Es gilt, die Space-Smilys zu befreien.

Space-Smilys sind kleine, gelbe, runde Dinger, die aussehen wie User-definierte Grafikzeichen (kein Wunder, da sie auch schließlich welche sind). Angeblich sind sie von irgendwoher aus dem Weltraum gekommen; jedenfalls ist eine Unmenge von ihnen in einem seltsamen Gangsystem mit direktem Ausgang in den Weltraum eingesperrt. Der Spieler steuert die Smilys mit den Tasten »Q«, »A«, »O« und »P«. Mit »Q« und »A« steuert man sie auf- und abwärts; mit »O« kann man ihren Vorwärtsdrang bremsen (sie bewegen sich immer vorwärts, es sei denn, eine Wand ist ihnen im Weg oder die Taste »O« ist gedrückt. Die Taste »P« wird benützt, um Space-Grumpys zu entfernen. Space-Grumpys sind im ganzen Gangsystem verteilt; man unterscheidet sie von den Space-Smilys, die immer lächeln (smile=lächeln), daß sie immer mürrisch sind (grumpy=mürrisch). Außerdem sind sie durch ihre ewig schlechte Laune schon magentafarben geworden. Space-Grumpys haben Space-Smilys zum Fressen gern (wortwörtlich zu nehmen). Berührt ein Smily einen Grumpy, so ist das Spiel zu Ende. Da sich die Grumpys aber vermehren (pro gerettetem Smily um 10) und die Gänge blockieren, können die Smilys, wenn sie direkt am Grumpy sind, diesen durch Druck auf Taste »P« in Staub und Asche umwandeln. Da die Grumpys manchmal auch Löcher in die Wände bohren und sich dann hineinsetzen, wird nach Entfernung eines solchen Grumpys ein neuer Durchgang frei, der oft angenehm als Abkürzung dient. Bei diesen Löchern in der Wand ist, solange die Grumpys noch darin sind, äußerste Vorsicht geboten! Die Smilys gehen nämlich auch dann hindurch und das Spiel endet. Mit Taste »O« kann dem vorgebeugt werden.

Außerdem liegen noch Käfige herum; wird einer von einem Smily berührt, so kommt dieses zum Ausgangspunkt zurück und es gibt 10 Minuspunkte.

Apropos Punkte: Pro gerettetem Smily gibt es 10, pro abgeschossenem Grumpy 5 Punkte.

Im günstigsten Fall dauert ein Spiel 5 Minuten; dann stoppt die über CLOCK gesteuerte Uhr das Programm. Meistens aber wird das Spiel sicherlich früher enden; außer die Grumpys setzen sich an die günstigsten Stellen der Mauern.

Eingeben und Laden des Programmes

Beta Basic laden, auf KEYWORDS 1 — Modus achten. Programm lt. Listing eingeben. Achtung! Wenn in Print-Posten »ALTER«, »B« oder »CLOCK« steht, so ist dies mit einem Tastendruck im Grafikmodus einzugeben, die Leerräume entstehen von selbst.

Programm auf Kassette saven.

Vor erneutem Laden zuerst Beta Basic laden!

Umschreiben für 16-KByte-Spectrum oder Spectrum ohne Beta Basic

Da Beta Basic viel Speicherplatz braucht, ist das Programm in der Form des Listings für den 16-KByte-Spectrum nicht geeignet. Für alle 16-KByte-Spectrum-Besitzer und alle Nicht-Beta-Basic-Besitzer hier Anweisungen zum Umschreiben des Programms:

1. folgende Zeilen löschen:
70, 120, 130, 160, 280, 450, 470, 510, 830
2. Änderungen:
Zeile 520—550, 820, 840: alle »PROC PRINT1« ändern in »GO SUB 590«
Zeile 590: »DEF PROC« ändern in »REM«
Zeile 620: »END PROC« ändern in »RETURN«
Zeile 150,200,210,250,520,940: in den Print-Posten das »ALTER«, »B« und »CLOCK« als »A«, »B« und »C« im Grafikmodus eingeben.
Zeile 810: Das »ALTER TO PAPER 7« weglassen oder änden in »PAPER 7: CLS«: das »ALTER TO PAPER 0« weglassen oder ändern in »PAPER 0: CLS«. Beim Weglassen geht ein guter Effekt verloren, beim Ändern wird der Bildschirm gelöscht, was bei ALTER nicht der Fall ist.
3. neue Zeilen:

```
65 DEF FN t()=(PEEK 23672+256*PEEK
23673+65536*PEEK 23674)/50
160 POKE 23672,0: POKE 23673,0: POKE 23674,0
280 IF FN t()>300 THEN GO TO 820
510 POKE 23658,8
830 PRINT AT 0, 31-LEN STR$ FN 6();FN t()
```

(Anmerkung: In FN t() steht die Zeit in Sekunden)

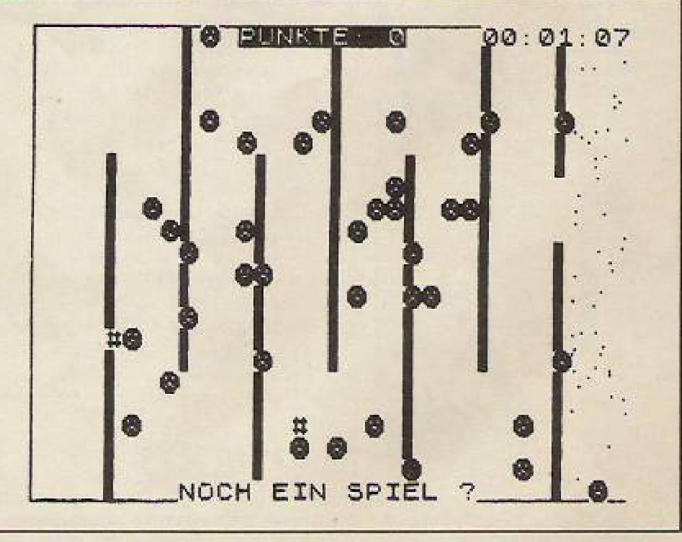
(H. Prillinger)

Variablenliste:

pts	Punktzahl
fi	1, wenn geschossen wird, sonst 0
sl,z	Zeilenzahl des Smilys
st,t	Spaltenzahl des Smilys
udg	Wert zum Poken der User-Grafics
a$	String, der in die Bildschirmmitte geschrieben wird
lin	Zeile, in die a$ geschrieben wird
k1,k2,k3	geben Auskunft über gedrückte Taste, siehe Zeile 650—670
e,f	z+e und t+f sind Position des abzuschießenden Grumpys
a,b	Steuervariablen von FOR/NEXT-Schleifen

Programmgliederung:

10 — 210	Initialisierung
220 — 290	Hauptschleife
300 — 380	Bildschirm
390 — 440	User-Grafics
450 — 470	wird aufgerufen, wenn Beta Basic fehlt
480 — 580	Programmerklärung
590 — 620	Printroutine
630 — 730	Tastaturabfrage
740 — 790	Attributabfrage
800 — 870	Crash
880 — 960	Grumpy abschießen



Hardcopy des Spielfeldes

```
   10 REM SPACE-SMILY
   20 REM VON HORST PRILLINGER
   30 REM JAHN-STRASSE 2
   40 REM A-5280 BRAUNAU AM INN
   50 REM ZX SPECTRUM BETABASIC
   60 REM
   70 IF PEEK 23731>234 THEN GO T
O 450
   80 GO SUB 390
   90 GO SUB 480
  100 LET PTS=0
  110 LET FI=0
  120C"00:00:00"
  130 KEYWORDS 0
  140 GO SUB 300
  150 FOR A=1 TO 30  PRINT AT INT
 (RND*22),INT (RND*26)+5; INK 3;
 BRIGHT 1;"B": NEXT A
  160C1
  170 LET SL=20: LET ST=1
  180 LET Z=20: LET T=1
  190 FOR A=1 TO 2: PRINT AT RND*
21,RND*31; INK 7;"#": NEXT A
  200 FOR A=1 TO 10: PRINT AT INT
 (RND*22),INT (RND*26)+5; INK 3;
 BRIGHT 1;"B": NEXT A
  210 PRINT AT Z,T;"A"
  220 IF IN 57342<>252 OR IN 5734
2<>253 OR IN 64510<>255 OR IN 65
022<>255 THEN GO SUB 630
  230 IF FI THEN GO SUB 880
  240 GO SUB 740
  250 PRINT AT Z,T;"A"
  260 IF PTS<0 THEN LET PTS=0
  270 IF T=31 THEN PRINT AT Z,T;"
 ": LET PTS=PTS+10: GO TO 170
  280 IF FN T$()="00:05:00" THEN
GO TO 820
  290 GO TO 220
  300 REM DISPLAY
  310 CLS
  320 FOR A=0 TO 6: PRINT AT A,28
; INK 4;"▌": NEXT A
  330 FOR A=10 TO 21: PRINT AT A,
28; INK 4;"▌": NEXT A
  340 FOR A=6 TO 21: PRINT AT A,4
; INK 4;"▌";AT A,12; INK 4;"▌";A
T A,20; INK 4;"▌": NEXT A
  350 FOR A=0 TO 15: PRINT AT A,8
; INK 4;"▌";AT A,16; INK 4;"▌";A
T A,24; INK 4;"▌": NEXT A
  360 INK 4: PLOT 0,0: DRAW 0,175
: DRAW 255,0: PLOT 0,0: DRAW 255
,0: INK 6
  370 FOR A=1 TO 50: PLOT INT (RN
D*24)+232,INT (RND*175): NEXT A
  380 RETURN
  390 REM UDG
  400 RESTORE 430
  410 FOR A=USR "A" TO USR "C"+7
  420 READ UDG: POKE A,UDG: NEXT
A
  430 DATA 60,126,219,255,219,231
,126,60,60,126,219,255,231,219,1
26,60,20,64,9,160,5,144,2,40
  440 RETURN
  450 CLS : PRINT AT 6,0;"FUER DI
ESES PROGRAMM IST"'"BETA BASIC V
ERSION 1.0"'"ERFORDERLICH."
  460 PRINT AT 12,0;"LADEN SIE ZU
ERST BETA BASIC,DANNERST DIESES
PROGRAMM."
  470 PRINT AT 16,0;"DRUECKEN SIE
 EINE TASTE.": PAUSE 0: NEW
  480 REM ERKLAERUNG
  490 BORDER 0: PAPER 0: INK 6: B
RIGHT 0: FLASH 0: OVER 0: INVERS
E 0: RANDOMIZE : CLS
  500 PRINT "
                    S P A C E - S
M I L Y
                 © 1984 HORST PR
ILLINGER"
  510 POKE 23658,8: KEYWORDS 0
  520 LET LIN=5: LET A$="STEUERN
SIE DIE SPACE-SMILYS (A)":■PRINT
1: LET A$="DURCH DIE KORRIDORE H
INAUS INS":■PRINT1: LET A$="ALL.
 ABER SPACE-SMILYS HABEN":■PRINT
1: LET A$="FEINDE (B), DIE GEFAE
HRLICH WER-":■PRINT1: LET A$="DE
N KOENNEN, WENN DIE SMILYS":■PRI
NT1: LET A$="NICHTS GEGEN SIE UN
TERNEHMEN.":■PRINT1
  530 LET A$="VORSICHT! SPACE-SMI
LYS SCHLUEP-":■PRINT1: LET A$="F
EN DURCH JEDES LOCH, AUCH":■PRIN
T1: LET A$="WENN EIN FEIND DARIN
 IST!":■PRINT1
  540 LET A$="WIRD EIN KAEFIG (#)
 BERUEHRT,":■PRINT1: LET A$="SO
GIBT ES PUNKTEABZUEGE UND":■PRIN
T1: LET A$="DAS SMILY MUSS ZUM A
USGANGS-":■PRINT1: LET A$="PUNKT
 ZURUECK.":■PRINT1
  550 LET LIN=LIN+1: LET A$="TAST
EN: Q-HINAUF  A-HINUNTER":■PRINT
1: LET A$="O-NICHT NACH RECHTS G
EHEN":■PRINT1: LET A$="P-FEIND A
BSCHIESSEN":■PRINT1
  560 PRINT #0;AT 1,8;"START MIT
ENTER."
  570 IF INKEY$<>CHR$ 13 THEN GO
TO 570
  580 RETURN
  590 ■PRINT1
  600 PRINT AT LIN,INT ((32-LEN A
$)/2);A$
  610 LET LIN=LIN+1
  620■
  630 REM TASTATUR
  640 LET K1=IN 57342: LET K2=IN
64510: LET K3=IN 65022
  650 REM P:K1=254/O:K1=253
  660 REM O+P:K1=252
  670 REM Q:K2=254/A:K3=254
  680 IF K2=254 THEN LET SL=SL-1
  690 IF K3=254 THEN LET SL=SL+1
  700 IF K1=254 OR K1=252 THEN LE
T FI=1
  710 IF (K1<>253 AND K1<>252) TH
EN LET ST=ST+1
  720 IF ATTR (SL,ST)=4 THEN LET
ST=ST-1
  730 RETURN
  740 REM FELD
  750 IF ATTR (SL,ST)=7 THEN LET
PTS=PTS-10: BEEP .02,40: PRINT A
T Z,T;" ": LET Z=20: LET T=1: LE
T SL=Z: LET ST=T: GO TO 210
  760 IF ATTR (SL,ST)<>6 AND ATTR
 (SL,ST)<>67 THEN BEEP .1,0: LET
 SL=Z: LET ST=T
  770 IF ATTR (SL,ST)=6 THEN BEEP
 .01,T-5: PRINT AT Z,T;" ": LET
Z=SL: LET T=ST
  780 IF ATTR (SL,ST)=67 THEN GO
TO 800
  790 RETURN
  800 REM CRASH
  810 FOR B=12 TO 30 STEP 6: FOR
A=30+B TO 10+B STEP -1:A TO PAPE
R 7:A TO PAPER 0: BEEP .001,A: N
EXT A: NEXT B
  820 INVERSE 1: LET LIN=0: LET A
$="PUNKTE: "+STR$ PTS:■PRINT1: I
NVERSE 0
  830 PRINT AT 0,24;FN T$():C0
  840 LET LIN=21: LET A$="NOCH EI
N SPIEL ?":■PRINT1: PAUSE 0
  850 IF INKEY$<>"N" AND INKEY$<>
"J" AND INKEY$<>"Y" THEN GO TO 8
50
  860 IF INKEY$="N" THEN STOP
  870 GO TO 100
  880 REM FIRE
  890 LET E=0: LET F=0
  900 IF ATTR (Z+1,T)=67 THEN LET
 E=1: LET F=0
  910 IF ATTR (Z-1,T)=67 THEN LET
 E=-1: LET F=0
  920 IF ATTR (Z,T+1)=67 THEN LET
 E=0: LET F=1
  930 IF ATTR (Z+E,T+F)=67 THEN L
ET PTS=PTS+5
  940 PRINT AT Z+E,T+F; INK 6;"C"
: FOR A=30 TO 20 STEP -1: BEEP .
003,A: NEXT A
  950 LET FI=0
  960 RETURN
```

Basic-Listing »Space-Smily«

Dreidimensionales Plotten

Hier wird erklärt, wie mathematische Funktionen mit dreidimensionalem Effekt dargestellt werden können. Das abgedruckte Programm läuft auf jedem Spectrum, ist aber auch für Benutzer anderer grafikfähiger Computer interessant.

Echte dreidimensionale Grafiken sind auf einem Bildschirm nur mit Spezialbrillen möglich. Wir beschränken uns hier auf eine perspektivische Abbildung. Da die mathematischen Funktionen in drei Achsen dargestellt werden, ist die Bezeichnung »dreidimensional« dennoch gerechtfertigt.

Ein Algorithmus zur zweidimensionalen Funktions-Darstellung ist recht einfach. Die Zahl der vertikalen und horizontalen Punkte ist geräteabhängig. Beim Spectrum sind 176*256 Punkte mit dem Basic-Befehl Plot setz- und löschbar.

Man kann eine Funktion natürlich nicht vollständig zeichnen — die Achsen sind ja unendlich lang —, sondern nur einen Ausschnitt. Dieser Bereich wird mit den Variablen xlinks, xrechts, max und min festgelegt. Die Extrema (Maximum und Minimum) können entweder von vornherein bekannt sein oder sie können von einem Programmteil durch »Ausprobieren« ermittelt werden.

Nun können die Funktionswerte berechnet und die entsprechenden Punkte der Reihe nach gesetzt werden. Die Position des zu setzenden Punktes errechnet sich aus der Formel

$$\text{Bildschirmhöhe} \cdot \frac{f(x) - \min}{\max - \min}$$

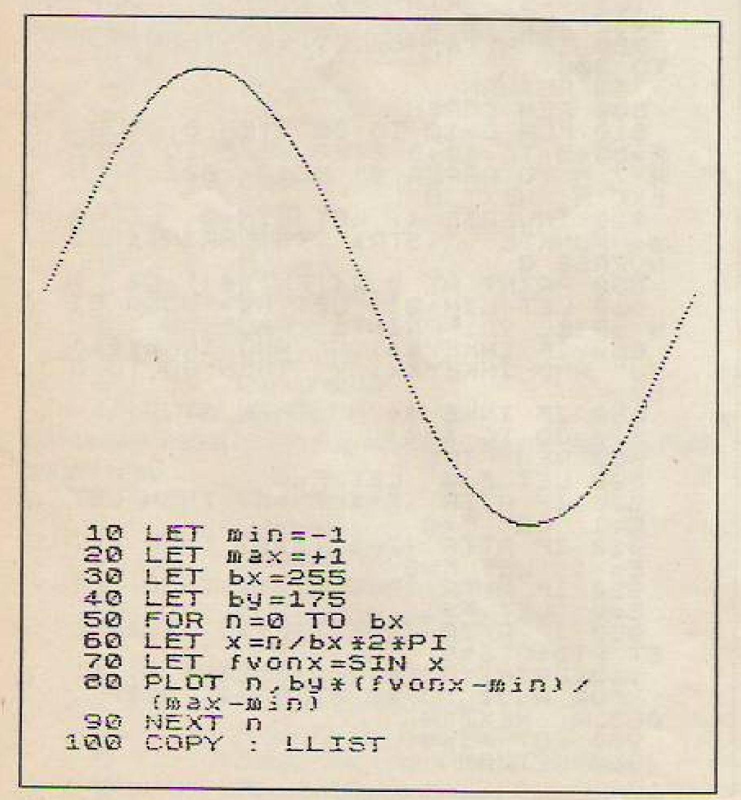


Bild 1. Eine Kurve hat zwei Dimensionen

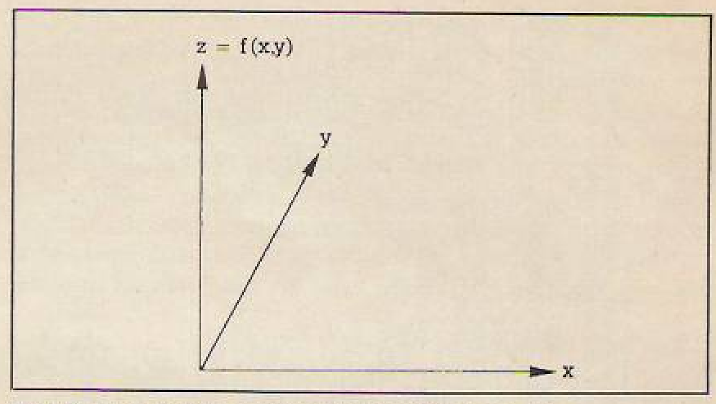


Bild 2. Das dreidimensionale Koordinatensystem

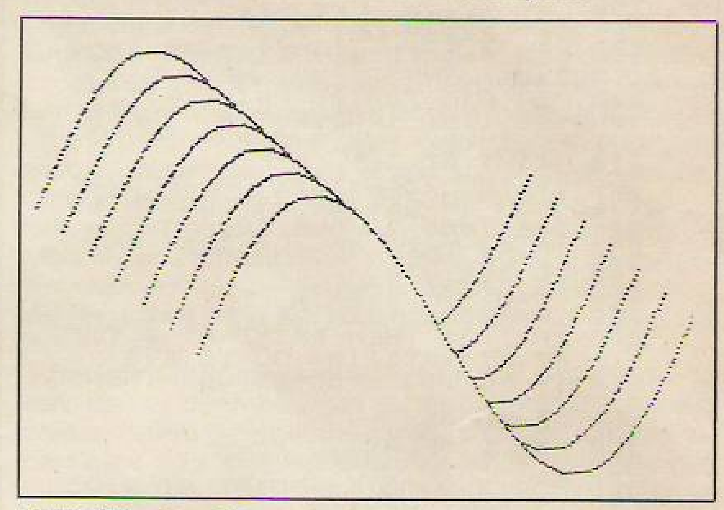


Bild 3. Eine Anzahl verschobener Kurven

Auf diese Weise entstand Bild 1. Zeile 100 kopiert den Bildschirminhalt auf den Drucker und listet das Programm auf.

Man kann entweder alle Positionen berechnen, in einem Feld abspeichern und erst zum Schluß zeichnen oder aber nach jeder Rechnung den entsprechenden Punkt sichtbar machen. Das zweite Verfahren hat den Vorteil, daß etwas auf dem Bildschirm passiert und die unvermeidliche Wartezeit nicht so lang erscheint. Außerdem wird weniger Speicherplatz gebraucht.

Wie entsteht nun die räumliche Wirkung der Bilder? Zu der waagrechten x-Achse und der senkrechten f(x)-Achse kommt eine weitere für die Tiefe. Dieses Koordinatensystem ist in Bild 2 skizziert. Die Funktion hängt außer von x jetzt auch noch von y ab, $z = f(x,y)$.

Der plastische Effekt tritt ein, wenn eine Reihe von Kurven jeweils leicht verschoben aufeinander gelegt wird (Bild 3). In der Projektion verdeckte Flächen, also das Innere des Funktions-Gebirges, soll nicht erscheinen. Man kann dieses Problem ignorieren, der Graph wirkt dann jedoch wie ein durchsichtiges Netz und nicht wie ein massiver Körper.

Beim hier besprochenen Programm wird das Bild in lauter Schichten von hinten nach vorn aufgebaut. Jede einzelne Schicht wird nicht als Kurve gezeichnet, sondern als Fläche, von der nur die obere Kante zu sehen ist. Das läßt sich mit einer Reihe ausgesägter Bretter vergleichen, die hintereinander gestellt werden. In Bild 3 ist diese Struktur deutlich zu erkennen.

Werden die Scheiben abwechselnd schwarz und weiß eingefärbt, entsteht der Streifen-Effekt von Bild 4. Der allmähliche Aufbau läßt sich am Bildschirm verfolgen. Dabei läßt sich die Arbeitsweise des Programms leicht nachvollziehen.

Sehen wir uns Bild 5 an. Es handelt sich um eine gedämpfte Schwingung, die um den Mittelpunkt rotiert. Eine gedämpfte Schwingung entsteht beispielsweise durch ein Pendel; der Ausschlag wird immer schwächer bis zum Stillstand. Die Rotation sieht schwieriger aus als sie ist. Die einfachere Funktion,

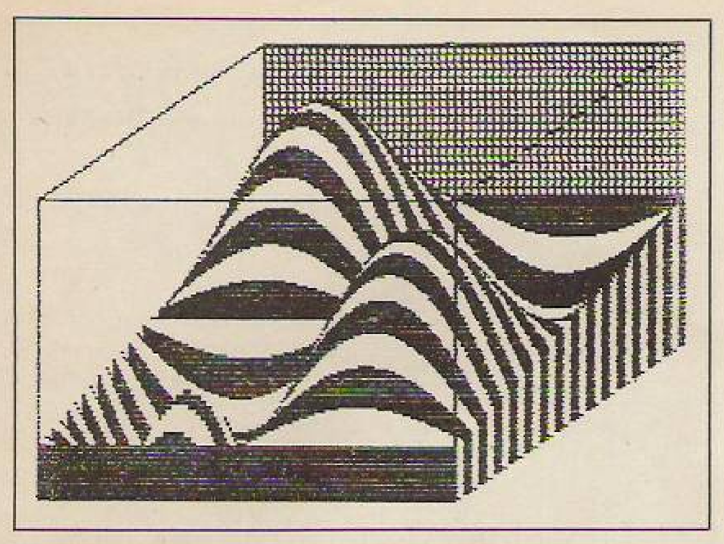


Bild 4. Die gestreiften Sinusberge

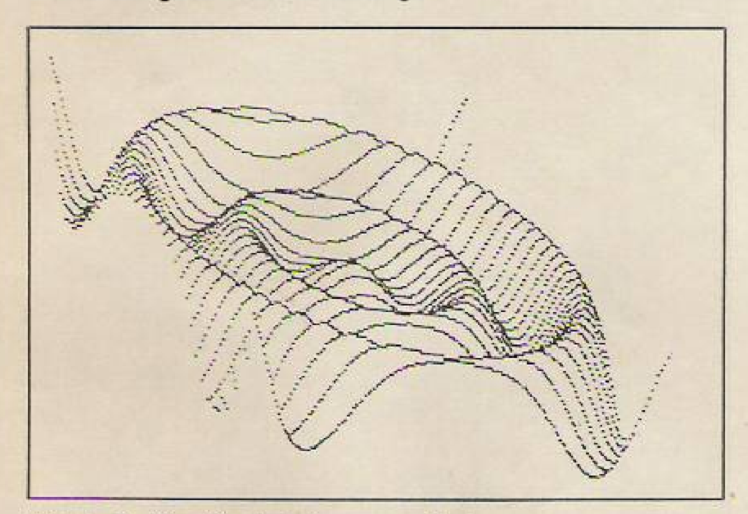


Bild 5. Gedämpfte Schwingung rotiert

wie die Sinusberge auf Bild 4, sind von x und y abhängig, also z=f(x,y). Eine rotierte Funktion ist jedoch abhängig vom Abstand zum Mittelpunkt.

Dieser Abstand läßt sich mit dem Satz des Pythagoras berechnen. In einem rechtwinkligen Dreieck ist die Summe der Kathetenquadrate gleich dem Hypotenusenquadrat (Bild 6). Die Seiten am rechten Winkel entsprechen hier x und y. Der Abstand zum Mittelpunkt ergibt sich somit aus

$$r^2 = x^2 + y^2$$

$$r = \sqrt{x^2 + y^2}$$

Die Zeile LET r = SQR (x*x+y*y) findet sich mit den Nummern 790 und 980 im Programm wieder. Wenn als Funktion nun zum Beispiel SIN r festgelegt wird, entsteht eine rotierte Sinuskurve.

Auch der »Hut« auf Bild 7 macht von der Rotation Gebrauch, ebenso Bild 8. Dieses Bild stellt eine plattgedrückte Kugel dar, die mit den Buchstaben »3D« beschriftet ist. Dabei wurde eine Besonderheit des Spectrum ausgenutzt: Die einzelnen Punkte eines mit PRINT geschriebenen Textes lassen sich mit der Funktion POINT abfragen. Mit PRINT AT 21,0;»3D« wurden die Buchstaben auf die linke untere Ecke gebracht. Beim Plotten wurden die Punkte dann abgefragt und ein Teil des Bildes dadurch leergelassen.

Am schwierigsten war die Pyramide auf Bild 9 festzulegen. Für die vier schrägen Flächen und den Rand ist eine umfangreiche Fallunterscheidung nötig, die nicht von IFs, sondern von AND-Operatoren geleistet wird.

Bild 10 zeigt die Gaußsche Normalverteilung. Diese Kurve hat in der Statistik eine große Bedeutung. Wenn Sie hundert Hasen fangen und die Häufigkeit ihrer Ohrenlänge auf Millimeterpapier eintragen, wird eine ähnliche Kurve herauskommen. Am Gipfel schlagen sich die zahlreichen Durchschnitts-Hasen nieder. Links und rechts sind die seltenen Tiere mit besonders kleinen beziehungsweise besonders großen Ohren berücksichtigt. Weil sie so selten sind, ist die Kurve dort niedriger.

In Bild 11 ist das Listing zu sehen. Das Basic-Programm ist in mehrere Blöcke unterteilt, die nacheinander durchlaufen werden. Manche halten sowas für unmöglich, aber es sind weder GOTOs noch GOSUBs in diesem Basic-Programm enthalten!

Es ist nicht einfach, die GOTOs zu vermeiden. Einige Programmteile wie zum Beispiel Gitter I dürfen nur unter bestimmten Bedingungen ausgeführt werden. In der Programmiersprache Pascal würde dieser Abschnitt lauten

```
IF gitter
THEN BEGIN
...
END;
```

In Basic hingegen darf auf ein THEN nur eine einzelne Zeile folgen. Darum würde an einer solchen Stelle überlicherweise ein GOTO benutzt: IF gitter = 0 THEN GOTO 700
Sprünge sind jedoch aus gutem Grund verpönt, denn sie zerstören die Programmstruktur. Man kann einfach nicht mehr folgen.

Der Autor hat eine solche Abneigung gegen GOTOs, daß er statt dessen FOR-NEXT-Schleifen benutzt. Im Beispiel werden die Zeilen 520-590 nur dann ausgeführt, wenn die Variable »gitter« ungleich Null ist; sonst nicht. Das klappt nur bei Sinclair-Geräten, in anderen Basic-Dialekten wird eine Schleife immer mindestens einmal durchlaufen.

Am Programm-Anfang werden eine Minimum- und eine Maximum-Funktion definiert. Sie benötigen zwei Parameter und liefern als Ergebnis den kleineren beziehungsweise den größeren Wert der beiden Zahlen. Dann erfolgen eine Reihe von Eingaben, sie werden naher noch besprochen.

Im Block Kalkulation werden die Variablen »ebenen«, »stepx« und »stepy« berechnet. Gitter I und II zeichnen den Kasten rund um die Funktionsdarstellung. Sie werden nur durchlaufen, wenn das vorher vereinbart wurde (»gitter« ungleich Null).

Auch der Block min/max-Automatik wird nur auf ausdrücklichen Wunsch benutzt (wenn »min« und »max« beide Null sind). Der Teil PLOT ist der wichtigste. Hier angekommen, stehen alle Variablen fest. Die Funktionswerte können nacheinander berechnet und gezeichnet werden.

Hinweise zum Eintippen

Beim Erreichen der Zeile 1200 ist das Bild fertig. Wenn ein Drucker angeschlossen ist, wird ausgedruckt. Alle wichtigen Parameter werden mit ausgegeben, damit man die Grafik später rekonstruieren kann.

Wer ohne langes Eingeben zu einem schnellem Ergebnis kommen will, kann die Programmteile Gitter I und II (Zeilen

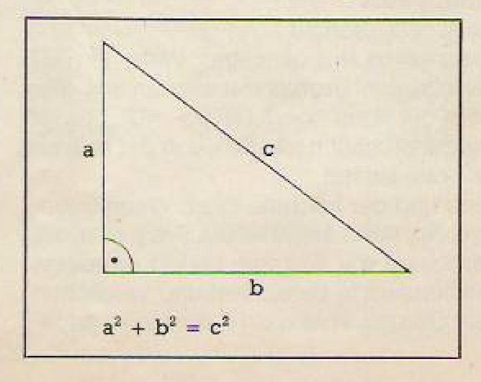


Bild 6. Der Satz des Pythagoras

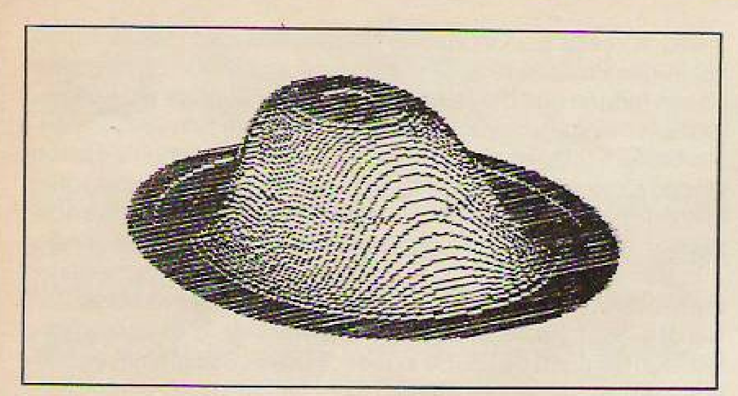


Bild 7. Hut

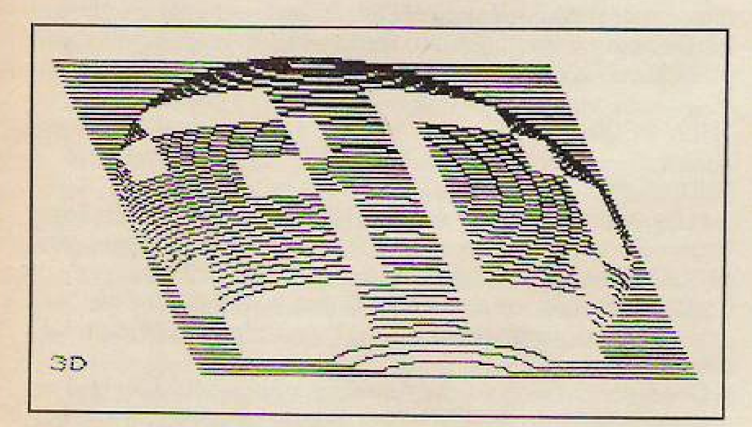


Bild 8. Eine plattgedrückte Kugel

500-600 sowie 1100-1170) und min/max-Automatik (Zeilen 700-830) erst einmal weglassen. Das Gitter wird dann eben nicht gezeichnet und Minimum/Maximum sollten bekannt sein.

Im Listing läßt sich die Null nur schwer vom Buchstaben »0« unterscheiden. Dennoch dürfte es nicht zu Mißverständnissen kommen: Der Buchstabe taucht nur innerhalb von Wörtern auf, alles übrige sind Nullen.

Die Funktionsnamen wie SIN oder EXP erscheinen als Ganzes im »Extended Mode« (Caps Shift und Symbol Shift gleichzeitig drücken, blinkendes E erscheint). Werden sie fälschlicherweise aus Einzelbuchstaben zusammengesetzt, reagiert der Spectrum mit »Nonsense in Basic« beziehungsweise blinkendem Fragezeichen-Cursor.

Die Bedienung

Nach dem Start mit RUN werden eine Reihe von Eingaben verlangt:

f(x,y)= Die darzustellende Funktion. Als Parameter sind x, y und r (r wie Radius, der Abstand vom Ursprung) möglich. Alle Basic-Funktionen, auch selbstdefinierte, dürfen vorkommen. Funktionsnamen nicht aus Einzelbuchstaben zusammensetzen.

xlinks, xrechts, yhinten, yvorn
der abzubildende Funktionsausschnitt

nicht definiert: Normalerweise Null eintasten. Wenn ein Teil des angegebenen Intervalls nicht gezeichnet werden soll, diesen Teil angeben, etwa in der Form "x<7 OR y>=0". Das ist zum Beispiel bei der Tangensfunktion nützlich, von der sich einige Werte nicht berechnen lassen.

min, max Der niedrigste und der höchste Wert. Wenn unbekannt, jeweils Null eingeben, dann ermittelt das Programm sie automatisch. Die Bestimmung der Extrema dauert allerdings eine Weile, da alle Funktionswerte berechnet und verglichen werden. Während dieser Operation wird ein Countdown angezeigt.

```
  10 REM     ####################
  20 REM     #  3 D   P L O T  #
  30 REM     ####################
  40 REM      mit ZX Spectrum
  50 REM
  60 REM
  70 REM von Oliver Voelckers
  80 REM      Brombeerweg 5
  90 REM      4500 Osnabrueck
 100 REM
 110 REM Minimumfunktion
 120 DEF FN m(a,b)=(a AND a<=b)+
(b AND a>b)
 130 REM Maximumfunktion
 140 DEF FN n(a,b)=(a AND a>=b)+
(b AND a<b)
 200 REM ##### Eingaben
 210 INPUT "f(x,y)=";f$'"xlinks=
 ";xlinks'"xrechts= ";xrechts'"y
hinten= ";              yhinten'"yv
orn= ";yvorn'"nicht definiert: "
;n$'"min= ";min'"max= ";max
 220 INPUT "fx= ";fx,"fy= ";fy'"
movex= ";movex,"movey= ";movey'"
gestreift? 0/1",        gestreift'"
Gitter? 0/1",gitter
 230 IF gitter THEN INPUT "gitte
rstep?",gitterstep
 300 REM ##### Bildschirmhoehe u
nd -breite
 310 LET bx=255
 320 LET by=175
 400 REM ##### Kalkulation
 410 IF NOT movex AND NOT movey
THEN INPUT "Ebenen?",ebenen
 420 IF NOT movex AND movey THEN
 LET ebenen=(by-fy)/ABS movey
 430 IF movex AND NOT movey THEN
 LET ebenen=(bx-fx)/ABS movex
 440 IF movex AND movey THEN LET
 ebenen=INT FN m((bx-fx)/ABS mov
ex,(by-fy)/ABS           movey)
 450 LET stepx=(xrechts-xlinks)/
fx
 460 LET stepy=(yvorn-yhinten)/(
ebenen+(ebenen=0))
 470 LET inv=NOT gestreift
 480 CLS
 500 REM ##### Gitter I
 510 FOR a=1 TO gitter<>0
 520 LET yfenster=by-fy AND move
y<0
 530 LET xfenster=bx-fx AND move
x<0
 540 FOR m=yfenster TO yfenster+
fy STEP gitterstep: PLOT xfenste
r,m: DRAW fx,0:           NEXT m
 550 FOR m=xfenster TO xfenster+
fx STEP gitterstep: PLOT m,yfens
ter: DRAW 0,fy:           NEXT m
 560 PLOT xfenster,yfenster: DRA
W 0,fy: DRAW ebenen*movex,ebenen
*movey
 570 PLOT xfenster,yfenster+fy:
DRAW fx,0: DRAW ebenen*movex,ebe
nen*movey
 580 PLOT xfenster+fx,yfenster+f
y: DRAW 0,-fy: DRAW ebenen*movex
,ebenen*movey
```

fx, fy Die Breite und Höhe jeder »Scheibe«. Der Bildschirm ist 255·175 Punkte groß. Die Scheibe muß etwas kleiner sein, weil jede zur vorherigen leicht versetzt wird. Brauchbare Werte sind zum Beispiel 165 und 85. Für eine Darstellung mit nur zwei Dimensionen — also nicht räumlich — 255 und 175 eingeben.

movex, movey Anzahl Punkte, um die jede Ebene horizontal und vertikal verschoben wird. Positive Werte für movex und negative für movey sorgen dafür, daß das Zeichnen links oben beginnt und rechts unten endet. Kleine Werte ergeben ein feineres und dichteres Bild, der Aufbau dauert aber auch länger. Guter Ergebnisse liefern 3 und —3. Für zweidimensionalen Plot beide Male mit Null antworten.

```
 590 PLOT xfenster+fx,yfenster:
DRAW -fx,0: DRAW ebenen*movex,eb
enen*movey
 600 NEXT a
 700 REM ##### min/max-Automatik
 710 FOR a=1 TO min=0 AND max=0
 720 LET min= 1.7 E 38
 730 LET max=-1.7 E 38
 740 FOR m=0 TO ebenen
 750 LET y=yhinten+m*stepy
 760 PRINT #1;AT 0,0;ebenen-m;TA
B 20;
 770 FOR n=0 TO fx
 780 LET x=xlinks+n*stepx
 790 LET r=SQR (x*x+y*y)
 800 IF NOT VAL n$ THEN LET z=VA
L f$: LET min=FN m(min,z): LET m
ax=FN n(max,z)
 810 NEXT n
 820 NEXT m
 830 NEXT a
 900 REM ##### Plot
 910 LET stauch=fy/(max-min)
 920 FOR m=0 TO ebenen
 930 LET xfenster=(bx-fx AND mov
ex<0)+m*movex
 940 LET yfenster=(by-fy AND mov
ey<0)+m*movey
 950 LET y=yhinten+m*stepy
 960 FOR n=0 TO fx
 970 LET x=xlinks+n*stepx
 980 LET r=SQR (x*x+y*y)
 990 IF NOT VAL n$ THEN LET z=IN
T (stauch*(VAL f$-min)): PLOT xf
enster+n,            yfenster+z:
 DRAW INVERSE inv;0,(fy AND move
y>0)-z: PLOT xfenster+n,yfenster
    +z
1000 NEXT n
1010 IF gestreift THEN LET inv=N
OT inv
1020 NEXT m
1100 REM ##### Gitter II
1110 FOR a=1 TO gitter<>0
1120 PLOT xfenster,yfenster: DRA
W 0,fy
1130 PLOT xfenster,yfenster+fy:
DRAW fx,0
1140 PLOT xfenster+fx,yfenster+f
y: DRAW 0,-fy
1150 PLOT xfenster+fx,yfenster:
DRAW -fx,0
1160 PLOT xfenster+(fx AND movex
<0),yfenster+(fy AND movey<0): D
RAW -ebenen*           movex,-eben
en*movey
1170 NEXT a
1200 REM ##### Ausdrucken
1210 LPRINT "f(x,y)=";f$'"x link
s= ";xlinks'"x rechts= ";xrecht
s'"y hinten= ";          yhinten'"y
vorn=    ";yvorn'"nicht definiert
: ";n$'"min= ";min'"max= ";max
1220 LPRINT "fx= ";fx,"fy= ";fy'
"movex= ";movex,"movey= ";movey
1230 COPY
```

Bild 11. Ohne GOTOs: Das vollständige Listing

gestreift 0 für nein, 1 für ja. Bei 1 wird nach jeder Ebene die Vorder- und Hintergrundfarbe gewechselt.

Gitter 0 für nein, 1 für ja. Bei einer positiven Antwort wird noch mit »gitterstep« nach der Größe des Rasters gefragt, mit der die Hinterseite unterlegt wird. Ist kein Raster gewünscht, darauf 255 angeben.

Ebenen Diese Eingabe wird nur verlangt, wenn für movex und movey Null angegeben, also eine zweidimensionale Darstellung gewünscht wurde. Es ist dann möglich, mehrere Kurven direkt aufeinanderzulegen. In der Regel soll nur eine einzelne Kurve gezeichnet werden, dafür 1 eingeben.

Die Vielzahl von Eingaben macht das Programm universell, es sind völlig unterschiedliche Grafiken möglich. Andererseits

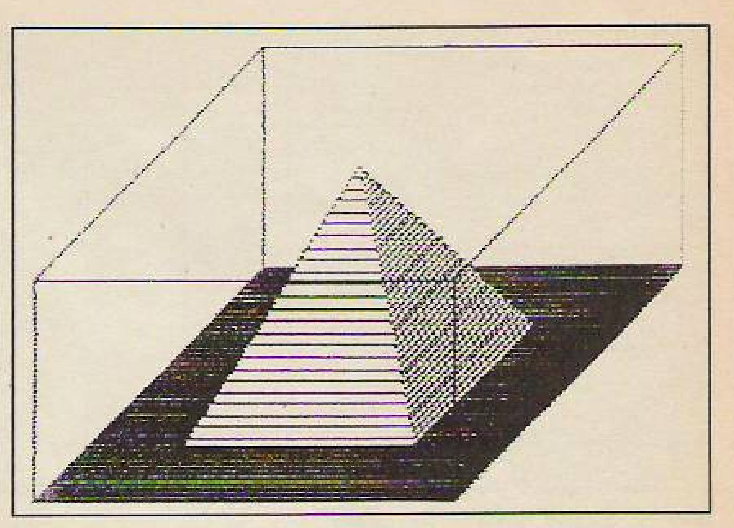


Bild 9. Pyramide

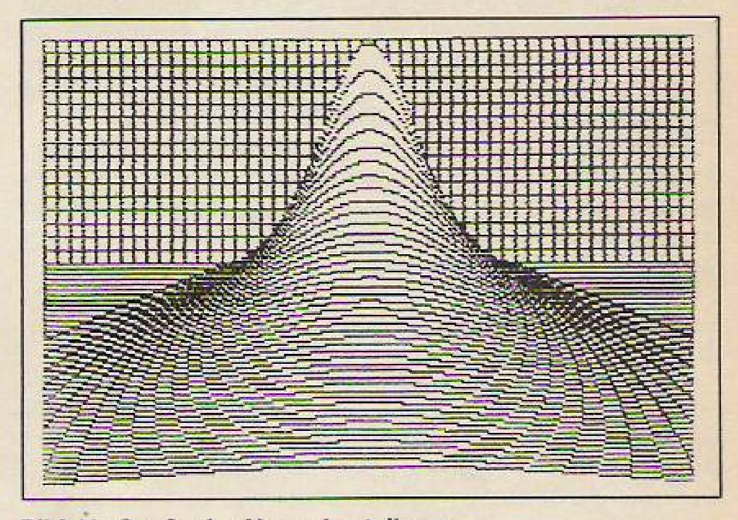


Bild 10. Gaußsche Normalverteilung

wird das Programm dadurch schwierig zu bedienen. Wer sich noch nicht damit auskennt, sollte erst nur die Funktion variieren (f(x,y), xlinks, xrechts, yhinten, yvorn, min, max) und für die übrigen Parameter die Standard-Werte benutzen.

Bild 12 zeigt eine Anzahl sinnvoller Eingaben, mit der die Bilder 4-10 gemacht wurden.

Bild 13 zeigt die verwendeten Variablen. In Bild 14 ist eine Erweiterung des Listings zu sehen. Werden diese Zeilen zusätzlich eingetippt und mit RUN 1300 gestartet, dann wird der Inhalt der DATA-Zeilen als Eingabe verwandt. Wenn diese Erweiterung läuft, malt und druckt der Spectrum ein Bild nach dem andern. Das Erstellen einer Grafik kann nämlich je nach Funktion zwischen fünf Minuten und mehreren Stunden dauern. Währenddessen kann man also ruhig Kaffee trinken gehen. Zeile 1330 enthält leider eine Sprunganweisung. Statt die Grafiken auszudrucken, können sie auch nacheinander auf Microdrive gespeichert werden. Zeile 1230 könnte dann lauten »SAVE*";1;f$(TO 4)SCREEN$«. Als Dateiname werden dann die ersten vier Buchstaben des Funktionsnamens genommen. Das Abspeichern dauert rund zehn Sekunden.

Übertragung auf andere Computer

Obwohl das Programm klar strukturiert ist, kann die Übertragung auf andere Computer erhebliche Schwierigkeiten bereiten. Das hängt damit zusammen, daß das Sinclair-Basic ungewöhnlich ist und andere Versionen nicht dasselbe können. Wer in dem kleinen Spectrum nur ein kleines Basic vermutet, wird sich wundern.

Funktion	xlinks	xrechts	yhinten	yvorn	nicht definiert	min	max	fx	fy	movex	movey	siehe Bild
SINx*SINy-0.1*y	0	2*PI	0	2*PI	0	-1	1	165	115	-3	-2	4
EXP(0.1*r)*COSr	-13	13	-13	13	0	-9	9	165	85	3	-3	5
SQR ABS(1-r*r) AND r<1	-1	1	1	-1	POINT((x+1)*7.5, (y+1)*4)	0	3	195	55	1	-2	8
(1-y AND x<=y AND x>-y) +(1-x AND x > y AND x>-y) +(y+1 AND x > y AND x<=-y) +(x+1 AND x<=y AND x<=-y) AND ABS x <=1 AND ABS y<=1	-1.8	1.8	-1.9	1.9	0	0	1	165	85	-1	-1	9
EXP(-0.5*x*x/y/y)/y/2.5	-3.5	3.5	0.5	3	0	0	0.81	255	85	0	-2	10

Bild 12. Einige Beispiel-Eingaben

Das fängt schon bei den ersten Zeilen an. Beim Spectrum dürfen selbstdefinierte Funktionen bis zu 26 Argumente haben, bei den meisten Microsoft-Versionen nur eines. Die Funktionen für Minimum und Maximum lassen sich dort also nicht verwenden und müssen umständlich ersetzt werden.

Die Funktionen AND, OR und NOT funktionieren in den verschiedenen Basic-Dialekten auch verschieden. Manche vergleichen die einzelnen Bits der internen Zahlendarstellung (zum Beispiel 6 AND 3 = 2). Andere, wie der Spectrum, prüfen nur, ob die Zahlen gleich oder ungleich Null sind. Der Spectrum behauptet, 6 AND 3 = 6, weil drei ungleich Null ist und somit 6 AND TRUE berechnet wird. Beide Verfahren haben ihre Vorteile.

Am nächsten Problem scheitern alle Geräte außer Sinclair und Acorn. Der Wert eines in einem String stehenden numerischen Ausdrucks soll berechnet werden. Beim Spectrum geht das mit VAL, beim Acorn mit EVAL. Andere Rechner schaffen es nicht, VAL"SIN (Z*3)" oder auch nur VAL"1+1" auszurechnen. Bei ihnen kann die VAL-Funktion nur einzelne Ziffern in eine Zahl umformen, etwa VAL"234".

Von der unterschiedlichen Interpretation der Schleifen war schon die Rede. Bei Nicht-Sinclair-Geräten müßte in den Zeilen 510, 710 und 1110 jeweils IF ... GOTO ... stehen.

Es versteht sich von selbst, daß ein Programm wie dieses nur auf einem grafikfähigen Computer ablaufen kann. Beispielsweise wäre es aussichtslos, das 3D-Programm auf einen normalen Commodore 64 zu übertragen, denn in dessen Basic kann man keine Punkte setzen geschweige denn Linien ziehen. Man brauchte dazu einen Toolkit oder eine Basic-Erweiterung.

Wie man sieht ist es fast unmöglich ein derartiges Basic-Programm geräteunabhängig zu schreiben. Natürlich kann man "3D-Plot" auch in der Programmiersprache Pascal formulieren. Dem steht im Wege, daß Grafik in Pascal nicht Standard ist. Außerdem gibt es kein Äquivalent zum Sinclair-VAL. Man müßte diese Funktion entweder selbst programmieren, das wäre aufwendig und langsam. Oder man müßte das Programm für jeden Durchgang neu kompilieren.

(Oliver Völkers)

a, b	Lokale Variablen für Minimum- und Maximum-Funktion
ebenen	Anzahl Schichten, mit denen die Grafik von hinten nach vorn aufgebaut wird.
fx, fy	Breite bzw. Höhe einer Schicht in Punkten
f$	die Funktion. Darf beliebig lang sein.
gestreift	0=nein, 1=ja
gitter	0=nein, 1=ja
gitterstep	Größe der Hinterseiten-Rasterung in Punkten
inv	Invers 0 (aus) oder 1 (ein). Zustand wird von Streifen zu Streifen gewechselt.
max	Höchster Punkt auf der z-Achse
min	Niedrigster Punkt auf der z-Achse
movex,movey	Verschiebung jeder Schicht nach rechts bzw. oben.
n$	numerischer Ausdruck, der angibt, welcher Bereich nicht gezeichnet werden soll.
stauch	Umrechnungsfaktor von der z-Achse in Bildschirmkoordinaten
stepx, stepy	Schrittweite auf der x- bzw. y-Achse, die der Schichten-Verschiebung entspricht.
xfenste, yfenster	Position der Ebene — des "Fensters" — auf dem Bildschirm
xlink,xrechts	Intervall auf der x-Achse
yhinten,yvorn	Intervall auf der y-Achse

◀ **Bild 13. Die verwendeten Variablen**

```
1240>PAUSE 500
1300 REM ##### Daten
1310 READ f$,xlinks,xrechts,yhin
ten,yvorn,n$,min,max
1320 READ fx,fy,movex,movey,gest
reift,gitter,gitterstep
1330 GO TO 300
1340 REM Sinus gestreift
1350 DATA "SIN x*SIN y-.1*y",0,P
I+PI,0,PI+PI,"0",-1,1
1360 DATA 255-90,175-60,-3,-2,1,
0,0
1370 REM gedaempfte Schwingung r
otiert
1380 DATA "EXP (.1*r)*COS r",-13
,13,-13,13,"0",-9,9
1390 DATA 165,85,3,-3,0,0,0
1400 REM Gaussche Normalverteilu
ng
1410 DATA "EXP (-.5*x*x/y/y)/y/2
.5",-3.5,3.5,.5,3,"0",0,.81
1420 DATA 255,85,0,-2,0,1,5
1430 REM Cosinus gedreht
1440 DATA "COS r",-10*PI,10*PI,-
10*PI,10*PI,"r>10*PI",-1,1
1450 DATA 255-30,175-120,-1,-4,0
,0,0
1460 REM gedaempfte Schwingung r
otiert II
1470 DATA "EXP (.1*(22-r))*COS r
",-22,22,-22,22,"r>22",-10,10
1480 DATA 255-90,175-90,-1,-1,0,
0,0
```

Bild 14. Das Erweiterungs-Listing für Stapelbetrieb

Super-schneller Z80-Disassembler

Mit dem hier veröffentlichten Disassembler, der in Maschinensprache geschrieben wurde, lassen sich Maschinenprogramme mit hoher Geschwindigkeit in die Assemblersprache übersetzen. Dieses Programm wurde auf einem Spectrum entwickelt, läßt sich aber mit kleinen Änderungen auf andere Computer mit Z80-Prozessoren übertragen.

Das komplette Maschinenprogramm, das sämtliche Z80-Befehle übersetzt, belegt 3213 Bytes ab der Adresse 25000. So haben Besitzer der 16-KByte-Version des Spectrum noch gut 4500 Bytes Speicherkapazität zur Verfügung, um ab der Adresse 28213 zu übersetzende Maschinenprogramme ablegen zu können (von der übrigbleibenden Speicherkapazität der 48-KByte-Version ganz zu schweigen). Mit dem Programm lassen sich auch Listings von Maschinenprogrammen erstellen, doch davon später.

Die Anwendung des Programms ist denkbar einfach. Das zugehörige Basic-Programm (Listing 2), über »LOAD ""« geladen, setzt in Zeile 100 den RAMTOP auf 24999 und lädt danach das Maschinenprogramm. Es meldet sich schließlich nach Drücken einer Taste mit der Frage nach der Startadresse. Man gibt nun die Adresse des ersten Bytes des zu übersetzenden Maschinenprogramms ein. Danach verlangt das Programm noch die Eingabe der Adresse des letzten Bytes. In Zeile 40 werden die Adressen 25002/3 und 25004/5 mit den eingegebenen Werten geladen. Daraufhin erscheint das übersetzte Maschinenprogramm auf dem Bildschirm (die Geschwindigkeit der Übersetzung ist vergleichbar mit »LIST«). Die linke Spalte gibt jeweils die Adresse des Befehls an (dezimal). In der 2. Spalte befinden sich die Hexcodes der Bytes des Befehls (1-4 Bytes; je nach Befehl). Die rechte Spalte zeigt die übersetzten Befehle, wobei alle Zahlen in dezimaler Form erscheinen. Relative Sprünge werden automatisch in absolute Adressen umgewandelt.

Aufbau-Hilfe

Das komplette Maschinenprogramm ist wie folgt aufgebaut: Die Adressen 25000-25037 enthalten Zwischenspeicher, die für den Programmablauf wichtig sind. In der Adresse 25000 befindet sich nach der Untersuchung eines Befehls die Byteanzahl dieses Befehls. Die Adressen 25002/3 enthalten während des Programmablaufs die Adresse des jeweils zu übersetzenden Befehls, während der Inhalt von 25004/5 nicht verändert wird, da er die Endadresse darstellt. Die 32 Bytes von 25006 bis 25037 entsprechen den ASCII-Codes der 32 Zeichen, die nach der Übersetzung eines Befehls auf dem Bildschirm oder Drucker ausgegeben werden.

Den größten Speicherplatz benötigt die Codierung sämtlicher Z80-Befehle, nämlich die 2434 folgenden Bytes von 25038 bis 27471. Im wesentlichen sind die Befehle so codiert, daß die ASCII-Codes der einzelnen Zeichen der Z80-Befehle hintereinander im Speicher abgelegt sind. Das Byte für das letzte Zeichen eines Befehls wird um 128 erhöht. Dieses Setzen des 7. Bits ermöglicht es dem Computer, dieses letzte Zeichen als solches zu erkennen. Außerdem kann er durch Abzählen der Bytes, die größer als 127 sind, die Adresse der Codierung eines bestimmten Befehls anhand seines Befehlscodes ermitteln, da die Codierungen nach Befehlscodes geordnet sind. Dazu ein Beispiel: »SCF« hat den Befehlscode 55. Der Computer zählt von 25039 beginnend nun 55 Bytes ab, bei denen das 7. Bit gesetzt ist, und kommt so zur Adresse 25279. 25280 enthält das erste Zeichen des Befehls: S=83; C=67; F=70+ 128 (letztes Zeichen). Eigentlich beginnt der Computer mit dem Zählen bei der Adresse 25038 und zählt dann 56 Bytes ab. Das ist sinnvoll, damit er für den Befehl »NOP« nicht 0 Bytes abzählen muß, was im Programm etwas umständlicher wäre. Auch die Befehle, die mit CB beziehungsweise ED beginnen, sind geordnet in einer Datei abgelegt und befinden sich direkt hinter den übrigen Befehlen. Zeichenketten, die bei vielen Befehlen gleich vorkommen wie zum Beispiel »LD«, »INC« oder »(HL)«, werden in einer gesonderten Datei zu einem Byte zusammengefaßt (zum Beispiel »LD«=92, »INC«=93, »(HL)«=91). Die Datei beginnt bei der Adresse 27319, enthält insgesamt 32 Codes und ist ähnlich wie die Hauptdatei aufgebaut. Für die Hauptdatei gibt es noch einige wichtige Spezialcodes:

0 : Befehl existiert nicht, zum Beispiel ED 3C
2 : CB
3 : ED
4 : DD
5 : FD
63 : unmittelbare Daten werden erwartet (»N«)
64 : relativer Sprung (Diskriminante wird erwartet)

Von der Adresse 27455 bis 27471 befinden sich die Zeichen des Kommentars »Befehl unmöglich«, der ausgegeben wird, wenn das Programm auf einen nicht existierenden Befehl stößt — zum Beispiel wenn das Programm in der Hauptdatei die Zahl 0 beziehungsweise 128 findet.

741 Bytes reichen aus

Und nun zum eigentlichen Maschinenprogramm (Listing 4), das nur 741 Bytes belegt und bei der Adresse 27472 beginnt. Hier beginnt die Hauptroutine, die die Disassemblierung aller Befehle steuert. Von (der Adresse) 27472 bis 27481 werden die 32 Bytes ab 25006 initialisiert (mit Spaces gefüllt). Von 27482 bis 27486 wird 25000 initialisiert (Anzahl der Bytes pro Befehl = 1 gesetzt). 27487 bis 27498 steuert das Erzeugen der Adresse des zu übersetzenden Befehls. Das Aufsuchen der Adresse der Codierung des jeweiligen Befehls wird von 27499 bis 27505 gesteuert. 27506 bis 27508 ruft eine Routine auf, die den Befehl im Hinblick auf unmittelbare Daten, Unmöglichkeit, CB, ED, DD, FD oder doppelte Codierung (zum Beispiel »LD«=92) untersucht, entsprechend verzweigt und erst zurückspringt, wenn der Befehl vollständig übersetzt worden ist oder festgestellt worden ist, daß der Befehl nicht existieren kann. In diesem Fall wird das Carry-Flag zurückgesetzt, sonst ist es beim Rücksprung gesetzt. Dieses Flag testet 27509 bis 27526 und erzeugt, wenn nötig, den Kommentar »Befehl unmöglich«. Die Routine für das Erzeugen der Hexcodes der zweiten Spalte (28020 bis 28062) wird von 27527 bis 27529 aufgerufen. Die Ausgabe der drei Spalten (Routine 28198 bis 28212) wird von 27530 bis 27532 aufgerufen. 27533 bis 27545 setzt 25002/3 auf die Adresse

COMPUTER-ZEITSCHRIFTEN

VON PROFIS FÜR PROFIS

COMPUTER PERSÖNLICH

Das aktuelle Fachmagazin für Personal-Computer.

- ★ Wenn Sie jetzt den Schritt vom Heim-Computer zur professionellen Anwendung eines Personal Computers planen
- ★ Wenn Sie beruflich oder privat bereits einen Personal Computer benutzen
- ★ Wenn Sie regelmäßig Informationen über das aktuelle Produktangebot benötigen
- ★ Wenn Sie selbst programmieren
- ★ Wenn Sie professionelle Hard- und Softwaretests suchen
- ★ Wenn Sie Ihr eigenes System möglichst effizient einsetzen wollen

dann ist »Computer persönlich«, das aktuelle Fachmagazin für Personal Computer, genau Ihre Zeitschrift.

Die konsequente Ausrichtung auf professionelle Anwendungen bietet Ihnen alle wichtigen Informationen.

Von Profis für Profis!

»Computer persönlich« gibt es alle 14 Tage neu bei Ihrem Zeitschriftenhändler oder im Computer-Fachgeschäft.

PC MAGAZIN

Einzige Wochenzeitung für Personal Computer im IBM-Standard.

Sie beschäftigen sich beruflich oder privat mit dem Einsatz und der Anwendung von Personal Computern?

Sie sind an aktuellen, professionellen Informationen über IBM-PCs, kompatible Systeme und deren professionellen Einsatz interessiert? Dann ist das PC Magazin genau auf Ihre persönlichen Bedürfnisse zugeschnitten.

Es wird von anerkannten und erfahrenen Fachjournalisten für professionelle Anwender und Fachleute geschrieben.

Es berichtet jede Woche ausschließlich über Computer im IBM-Standard und kompatible Systeme, über Hard- und Softwareneuheiten. Es bringt ausführliche Testberichte und gibt Ihnen wichtige Informationen über Netzwerke sowie die PC/Host-Verbindung.

Nur diese Spezialisierung ermöglicht eine gezielte Berichterstattung und bietet genügend Raum, um auf Anwenderprobleme spezifisch eingehen zu können.

Von Profis für Profis!

Und das jeden Mittwoch neu bei Ihrem Zeitschriftenhändler oder im Computer-Fachgeschäft.

GUTSCHEIN

für ein kostenloses Probeexemplar

Senden Sie mir die neueste Ausgabe der von mir angekreuzten Zeitschrift kostenlos als Probeexemplar:

☐ **COMPUTER PERSÖNLICH**
Wenn mir Computer persönlich zusagt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte Computer persönlich dann regelmäßig alle 14 Tage per Post frei Haus geliefert und bezahle pro Jahr nur DM 98,—. Zustellung und Postgebühren übernimmt der Verlag.

☐ **PC-MAGAZIN**
Wenn mir das PC-Magazin zusagt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte mein PC-Magazin dann regelmäßig jede Woche per Post frei Haus geliefert und bezahle pro Jahr nur DM 155,—. Zustellung und Postgebühren übernimmt der Verlag.

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Vorname/Name

Straße PLZ/Ort

Datum 1. Unterschrift

Datum 2. Unterschrift

Gutschein ausfüllen, ausschneiden, auf Postkarte kleben und einsenden an:

```
  10 CLEAR 24999: POKE 23609,25
  20 FOR p=25000 TO 25029: POKE
p,0: NEXT p
  30 FOR p=25030 TO 28210 STEP 1
0
  35 POKE 23658,8
  40 INPUT "Hexcodes : ";h$: IF
LEN h$<>20 THEN PRINT #0;"Falsch
eingabe!Bitte wiederholen!": PAU
SE 0: GO TO 35
  50 FOR n=0 TO 18 STEP 2
  60 LET x=CODE h$(n+1)-48: LET
y=CODE h$(n+2)-48
  65 IF x>22 OR y>22 THEN LET c=
1: GO TO 150
  70 IF x>9 THEN LET x=x-7
  80 IF y>9 THEN LET y=y-7
  90 POKE p+n/2,x*16+y
 100 NEXT n
 110 INPUT "Pruefsumme :",c
 120 FOR n=0 TO 9
 130 LET c=c-PEEK (p+n)
 140 NEXT n
 150 IF c THEN BEEP .5,0: PRINT
"Falsche Eingabe; Zeile bitte"'"
erneut eingeben!": PAUSE 0: CLS
: GO TO 35
 160 PRINT AT 0,0;"Zeile ";p;" k
orrekt eingegeben."
 170 NEXT p
 180 PRINT "Fertig."
```

Listing 1. Basic-Hilfsprogramm

des nächsten Befehls. Die zuletzt gedrückte Taste (beziehungsweise die Adresse 23560) wird von 27546 bis 27551 getestet. Wenn man die Spacetaste während des Programmablaufs drückt, stoppt das Programm sofort und startet neu — Eingabe von Start- und Endadresse und so weiter 27552 bis 27563 testet, ob die Endadresse schon erreicht ist und verzweigt entweder nach 27472 oder kehrt zum Basic-Programm zurück. Die Routine ab 27578 sucht die Adressen der doppelt codierten Zeichenketten und lädt diese Ketten in den Speicherbereich von 25006 bis 25037, der später ausgegeben wird. Nach 27699 wird verzweigt, wenn der untersuchte Befehl ein IX- oder IY-Befehl ist. Dort wird er dann weiter untersucht. Von der Routine ab 27955 werden die Zeichencodes für die indizierte Adressierung erzeugt (zum Beispiel »(IY-15)«) und in den Speicherbereich von 25006 bis 25037 geladen. Die Routine ab 28063 steuert das Ersetzen von unmittelbaren Daten (»N« oder »NN«) durch Zahlen. Das Erzeugen von Dezimalzahlen geschieht durch die Routine von 28108 bis 28197.

Eingabe-Hinweise

Der Ablauf bei der Eingabe des Programms empfiehlt sich wie folgt: Zunächst wird das Hilfsprogramm (Listing 1) für die Eingabe des Maschinenprogramms eingegeben und mit »RUN« gestartet. Nun muß man — möglichst sorgfältig — sämtliche Hexcode-Zeichenketten von Listing 3 hintereinander eingeben. Dafür ein kleiner Tip: Wenn Sie eine Zeichenkette eingeben, so muß das letzte Zeichen dieser Kette am Zeilenende und der Cursor am Anfang der nächsten Zeile stehen. Natürlich prüft das Programm auch selbständig die Länge des eingegebenen Strings; sollte sie aber nicht 20 betragen, so müssen Sie den gesamten String neu eingeben. Hinter jeder Reihe fordert das Programm die Prüfsumme der Reihe; diese befindet sich wie üblich jeweils rechts vom Pfeil. Wenn das Maschinenprogramm vollständig eingegeben ist, kann man das Hilfsprogramm mit »NEW« löschen und das Basic-Programm (Listing 2) eingeben. Abgespeichert wird das komplette Programm über »RUN 9990«. Überprüfen kann man das eingegebene Programm, indem man es mit »RUN« startet und als Startadresse 27472, als Endadresse zum Beispiel 30000 eingibt. Die ausgegebenen Daten kann man mit dem Listing 4 vergleichen.

```
   1 REM Z80-Disassembler von
        Axel Schimice
  10 BORDER 1: PAPER 1: INK 7: C
LS
  20 INPUT "Startadresse : ";s
  30 INPUT "Endadresse : ";e
  40 POKE 25002,s-INT (s/256)*25
6: POKE 25003,INT (s/256): POKE
25004,e-INT (e/256)*256: POKE 25
005,INT (e/256)
  50 RANDOMIZE USR 27472
  60 GO TO 20
 100 BORDER 2: PAPER 6: INK 1: C
LEAR 24999
 110 PRINT AT 5,5; BRIGHT 1; FLA
SH 1;"DISASSEMBLER Version 3"; F
LASH 0;AT 10,0;"  1984 by Axel S
chimice,Velbert "
 120 PRINT AT 15,0;: LOAD ""CODE

 130 PRINT #0;"Zum Weitermachen
Taste druecken!": PAUSE 0: RUN
9990 SAVE "Disass.III" LINE 100:
 SAVE "MC/DATA"CODE 25000,3213
```

Listing 2. Basic-Vorprogramm

Um Listings von Maschinenprogrammen zu erstellen, gibt man »POKE 28199,3« ein (normal: PEEK 28199=2). Dadurch wird der Drucker- und nicht der Bildschirmkanal geöffnet. Die Anzahl der Zeichen pro Zeile muß bei den meisten Druckern auf 32 gesetzt werden. Man kann natürlich auch eine eigene Ausgaberoutine anfertigen.

Umarbeitung möglich

Dies müssen auch Besitzer anderer Computertypen tun, da die Routinen 5633 und 8252 spectrumspezifische ROM-Routinen sind, die einen bestimmten Kanal öffnen beziehungsweise eine bestimmte Anzahl von Bytes (BC) ab einer bestimmten Adresse (DE) ausgeben. Außerdem müssen die 6 Bytes von 27546 bis 27551 für andere Computer auch NOPs ersetzt werden, da die Adresse 23560 ebenfalls eine spectrumspezifische Programmvariable darstellt, die den Code der zuletzt gedrückten Taste enthält. Das Umschreiben des Basic-Programms (Listing 2) dürfte nicht schwerfallen.

(Axel Schimice)

Das Listing 3. (»Hexcode«) und Listing 4. finden Sie ab Seite 60

```
25030 : 0000000000000000804E -> 206
25040 : 4FD05C42432C3FBF5C28 -> 942
25050 : 4243292CC15D42C35DC2 -> 1052
25060 : 5EC25C422CBF524C43C1 -> 1099
25070 : 45582041462C4146A741 -> 735
25080 : 444420484C2C42C35C41 -> 778
25090 : 2C284243A95E42C35DC3 -> 1029
25100 : 5EC35C432CBF525243C1 -> 1107
25110 : 444A4E5A20C05C44452C -> 807
25120 : 3FBF5C284445292CC15D -> 894
25130 : 44C55DC45EC45C442CBF -> 1239
25140 : 524CC14A5220C0414444 -> 932
25150 : 20484C2C44C55C412C28 -> 730
25160 : 4445A95E44C55DC55EC5 -> 1246
25170 : 5C452CBF5252C14A5220 -> 941
25180 : 4E5A2CC05C484C2C3FBF -> 942
25190 : 5C283F3F292C48CC5D48 -> 784
25200 : CC5DC85EC85C482CBF44 -> 1258
25210 : 41C14A52205A2CC04144 -> 905
25220 : 4420484C2C48CC5C484C -> 808
25230 : 2C283F3FA95E48CC5DCC -> 1046
25240 : 5ECC5C4C2CBF4350CC4A -> 1126
25250 : 52204E432CC05C53502C -> 794
25260 : 3FBF5C283F3F292CC15D -> 883
25270 : 53D05DDB5EDB5C5B2CBF -> 1334
25280 : 5343C64A5220432CC041 -> 904
25290 : 444420484C2C53D05C41 -> 808
25300 : 2C283F3FA95E53D05DC1 -> 1050
25310 : 5EC15C412CBF4343C65C -> 1103
25320 : 422CC25C422CC35C422C -> 903
25330 : C45C422CC55C422CC85C -> 1089
25340 : 422CCC5C422CDB5C422C -> 937
25350 : C15C432CC25C432CC35C -> 1080
25360 : 432CC45C432CC55C432C -> 910
25370 : C85C432CCC5C432CDB5C -> 1121
25380 : 432CC15C442CC25C442C -> 906
25390 : C35C442CC45C442CC55C -> 1088
25400 : 442CC85C442CCC5C442C -> 924
25410 : DB5C442CC15C452CC25C -> 1107
25420 : 452CC35C452CC45C452C -> 914
25430 : C55C452CC85C452CCC5C -> 1103
25440 : 452CDB5C452CC15C482C -> 938
25450 : C25C482CC35C482CC45C -> 1093
25460 : 482CC55C482CC85C482C -> 929
25470 : CC5C482CDB5C482CC15C -> 1124
25480 : 4C2CC25C4C2CC35C4C2C -> 933
25490 : C45C4C2CC55C4C2CC85C -> 1109
25500 : 4C2CCC5C4C2CDB5C4C2C -> 967
25510 : C15C5B2CC25C5B2CC35C -> 1128
25520 : 5B2CC45C5B2CC55C5B2C -> 982
25530 : C85C5B2CCC48414CD45C -> 1148
25540 : 5B2CC15C412CC25C412C -> 924
25550 : C35C412CC45C412CC55C -> 1082
25560 : 412CC85C412CCC5C412C -> 915
25570 : DB5C412CC15FC25FC35F -> 1287
25580 : C45FC55FC85FCC5FDB5F -> 1491
25590 : C160C260C360C460C560 -> 1455
25600 : C860CC60DB60C164C264 -> 1498
25610 : C364C464C564C864CC64 -> 1492
25620 : DB64C162C262C362C462 -> 1489
25630 : C562C862CC62DB62C169 -> 1510
25640 : C269C369C469C569C869 -> 1507
25650 : CC69DB69C16BC26BC36B -> 1536
25660 : C46BC56BC86BCC6BDB6B -> 1551
25670 : C16AC26AC36AC46AC56A -> 1505
25680 : C86ACC6ADB6AC168C268 -> 1536
25690 : C368C468C568C868CC68 -> 1512
25700 : DB68C16F4EDA6742C370 -> 1399
25710 : 4E5A2C3FBF703FBF654E -> 1011
25720 : 5A2C3FBF6642C35FBF52 -> 1119
25730 : 535420B06FDAEF705A2C -> 1189
25740 : 3FBF82655A2C3FBF653F -> 1037
25750 : BF60BF52535420B86F4E -> 1132
25760 : C36744C5704E432C3FBF -> 1118
25770 : 72283F292CC1654E432C -> 785
25780 : 3FBF6644C564BF525354 -> 1161
25790 : 2031B66FC34558D87043 -> 1121
25800 : 2C3FBF71412C283FA965 -> 893
25810 : 432C3FBF8462BF525354 -> 1035
25820 : 2032B46F50CF6748CC70 -> 1151
25830 : 504F2C3FBF4558202853 -> 769
25840 : 50292C48CC65504F2C3F -> 808
25850 : BF6648CC69BF52535420 -> 1146
25860 : 33B26F50C570DB705045 -> 1209
25870 : 2C3FBF45582044452C48 -> 740
25880 : CC6550452C3FBF836BBF -> 1181
25890 : 5253542034B06FD06741 -> 996
25900 : C670502C3FBF44C96550 -> 1138
25910 : 2C3FBF6641C66ABF5253 -> 1125
25920 : 542034B86FCD5C53502C -> 967
25930 : 48CC704D2C3FBF45C965 -> 1134
25940 : 4D2C3FBF8568BF525354 -> 1052
25950 : 2035B675C275C375C475 -> 1320
25960 : C575C875CC75DB75C177 -> 1600
25970 : C277C377C477C577C877 -> 1577
25980 : CC77DB77C174C274C374 -> 1591
25990 : C474C574C874CC74DB74 -> 1596
26000 : C176C276C376C476C576 -> 1565
26010 : C876CC76DB76C178C278 -> 1604
26020 : C378C478C578C878CC78 -> 1592
26030 : DB78C179C279C379C479 -> 1601
26040 : C579C879CC79DB79C180 -> 1625
26050 : 80808080808080807AC27A -> 1334
26060 : C37AC47AC57AC87ACC7A -> 1602
26070 : DB7AC16C302CC26C302C -> 1128
26080 : C36C302CC46C302CC56C -> 1096
26090 : 302CC86C302CCC6C302C -> 896
26100 : DB6C302CC16C312CC26C -> 1115
26110 : 312CC36C312CC46C312C -> 886
26120 : C56C312CC86C312CCC6C -> 1111
26130 : 312CDB6C312CC16C322C -> 908
26140 : C26C322CC36C322CC46C -> 1097
26150 : 322CC56C322CC86C322C -> 895
26160 : CC6C322CDB6C322CC16C -> 1128
26170 : 332CC26C332CC36C332C -> 890
26180 : C46C332CC56C332CC86C -> 1107
26190 : 332CCC6C332CDB6C332C -> 924
26200 : C16C342CC26C342CC36C -> 1098
26210 : 342CC46C342CC56C342C -> 897
26220 : C86C342CCC6C342CDB6C -> 1139
26230 : 342CC16C352CC26C352C -> 893
26240 : C36C352CC46C352CC56C -> 1106
26250 : 352CC86C352CCC6C352C -> 911
26260 : DB6C352CC16C362CC26C -> 1125
26270 : 362CC36C362CC46C362C -> 901
26280 : C56C362CC86C362CCC6C -> 1121
26290 : 362CDB6C362CC16C372C -> 923
26300 : C26C372CC36C372CC46C -> 1107
26310 : 372CC56C372CC86C372C -> 910
```

Listing 3. »Hexcode«

```
26320 : CC6C372CDB6C372CC16D -> 1139
26330 : 302CC26D302CC36D302C ->  883
26340 : C46D302CC56D302CC86D -> 1104
26350 : 302CCC6D302CDB6D302C ->  917
26360 : C16D312CC26D312CC36D -> 1095
26370 : 312CC46D312CC56D312C ->  890
26380 : C86D312CCC6D312CDB6D -> 1136
26390 : 312CC16D322CC26D322C ->  886
26400 : C36D322CC46D322CC56D -> 1103
26410 : 322CC86D322CCC6D322C ->  904
26420 : DB6D322CC16D332CC26D -> 1122
26430 : 332CC36D332CC46D332C ->  894
26440 : C56D332CC86D332CCC6D -> 1118
26450 : 332CDB6D332CC16D342C ->  916
26460 : C26D342CC36D342CC46D -> 1104
26470 : 342CC56D342CC86D342C ->  903
26480 : CC6D342CDB6D342CC16D -> 1135
26490 : 352CC26D352CC36D352C ->  898
26500 : C46D352CC56D352CC86D -> 1114
26510 : 352CCC6D352CDB6D352C ->  932
26520 : C16D362CC26D362CC36D -> 1105
26530 : 362CC46D362CC56D362C ->  905
26540 : C86D362CCC6D362CDB6D -> 1146
26550 : 362CC16D372CC26D372C ->  901
26560 : C36D372CC46D372CC56D -> 1113
26570 : 372CC86D372CCC6D372C ->  919
26580 : DB6D372CC16E302CC26E -> 1126
26590 : 302CC36E302CC46E302C ->  887
26600 : C56E302CC86E302CCC6E -> 1115
26610 : 302CDB6E302CC16E312C ->  909
26620 : C26E312CC36E312CC46E -> 1101
26630 : 312CC56E312CC86E312C ->  896
26640 : CC6E312CDB6E312CC16E -> 1132
26650 : 322CC26E322CC36E322C ->  891
26660 : C46E322CC56E322CC86E -> 1111
26670 : 322CCC6E322CDB6E322C ->  925
26680 : C16E332CC26E332CC36E -> 1102
26690 : 332CC46E332CC56E332C ->  898
26700 : C86E332CCC6E332CDB6E -> 1143
26710 : 332CC16E342CC26E342C ->  894
26720 : C36E342CC46E342CC56E -> 1110
26730 : 342CC86E342CCC6E342C ->  912
26740 : DB6E342CC16E352CC26E -> 1129
26750 : 352CC36E352CC46E352C ->  902
26760 : C56E352CC86E352CCC6E -> 1125
26770 : 352CDB6E352CC16E362C ->  924
26780 : C26E362CC36E362CC46E -> 1111
26790 : 362CC56E362CC86E362C ->  911
26800 : CC6E362CDB6E362CC16E -> 1142
26810 : 372CC26E372CC36E372C ->  906
26820 : C46E372CC56E372CC86E -> 1121
26830 : 372CCC6E372CDB6E372C ->  940
26840 : C180808080808080808080 -> 1345
26850 : 80808080808080808080 -> 1280
26860 : 80808080808080808080 -> 1280
26870 : 80808080808080808080 -> 1280
26880 : 80808080808080808080 -> 1280
26890 : 80808080808080808080 -> 1280
26900 : 80808080807142 2C2843 ->  970
26910 : A9722843292CC26342C3 -> 1029
26920 : 5C283F3F292C42C34E45 ->  751
26930 : C7524554CE494D20B05C -> 1090
```

```
26940 : 492CC171432C2843A972 ->  924
26950 : 2843292CC36142C35C42 ->  903
26960 : 432C283F3FA980524554 ->  809
26970 : C9805C522CC171442C28 -> 1005
26980 : 43A9722843292CC46344 ->  905
26990 : C55C283F3F292C44C580 ->  933
27000 : 80494D20B15C412CC971 -> 1002
27010 : 452C2843A9722843292C ->  695
27020 : C56144C55C44452C283F ->  935
27030 : 3FA98080494D20B25C41 -> 1005
27040 : 2CD271482C2843A97228 ->  913
27050 : 43292CC86348CC5C283F ->  922
27060 : 3F292C48CC8080805252 ->  972
27070 : C4714C2C2843A9722843 ->  926
27080 : 292CCC6148CC5C484C2C ->  946
27090 : 283F3FA9808080524CC4 -> 1073
27100 : 71462C2843A980635300 -> 1021
27110 : 5C283F3F292C53D08080 ->  890
27120 : 808071412C2843A97228 ->  908
27130 : 43292CC16153D05C5350 ->  988
27140 : 2C283F3FA98080808080 -> 1019
27150 : 80808080808080808080 -> 1280
27160 : 80808080808080808080 -> 1280
27170 : 80808080808080808080 -> 1280
27180 : 804C44C94350C9494EC9 -> 1173
27190 : 4F5554C98080808040C44 -> 1105
27200 : C44350C4494EC44F5554 -> 1134
27210 : C480808080 4C4449D243 -> 1202
27220 : 5049D2494E49D24F5449 -> 1033
27230 : D280808080 4C4444D243 -> 1211
27240 : 5044D2494E44D24F5444 -> 1018
27250 : D2808080808080808080 -> 1362
27260 : 80808080808080808080 -> 1280
27270 : 80808080808080808080 -> 1280
27280 : 80808080808080808080 -> 1280
27290 : 80808080808080808080 -> 1280
27300 : 80808080808080808080 -> 1280
27310 : 80808080808080808080 -> 1280
27320 : 28484CA94C44A0494E43 ->  879
27330 : A0444543A04144442041 ->  822
27340 : AC4144432041AC414443 ->  841
27350 : 20484CAC5342432041AC ->  837
27360 : 53424320484CAC535542 ->  802
27370 : A043414C4CA050555348 ->  924
27380 : A0504F50A04350A0414E -> 1009
27390 : 44A04F52A0584F52A042 -> 1024
27400 : 4954A0524553A0534554 ->  947
27410 : A0524554A04A50A0494E -> 1020
27420 : A04F5554A0525354A052 -> 1059
27430 : 4CA0524C43A05252A052 -> 1027
27440 : 5243A0534C41A0535241 ->  923
27450 : A053524CA04265666568 -> 1035
27460 : 6C20756E6D6F65676C69 -> 1004
27470 : 63E821AE610620787723 ->  947
27480 : 10FC3E0132A861DD21AE -> 1074
27490 : 612AAA61E5CDCC6DE146 -> 1448
27500 : 040E01CDAC6BCDD76B38 -> 1086
27510 : 10213F6B7ECBBFDD7700 -> 1079
27520 : CB7E23DD2328F3CD746D -> 1333
27530 : CD266E2AAA61ED4BA861 -> 1239
27540 : 06000922AA613A085CFE ->  728
27550 : 20C82AAA61ED5BAC61ED -> 1375
27560 : 5238A5C921CE61CB7E23 -> 1204
27570 : 28FB10F90D20F6C9D65A -> 1352
27580 : E521B76ACB7E2328FB3D -> 1267
27590 : 20F87EE67FDD7700DD23 -> 1359
27600 : CB7E2328F3E1C9DD21BD -> 1516
27610 : 617EE67FC8FE5B3805CD -> 1391
27620 : BA6B1844FE063014FE04 ->  971
27630 : 30432AAA61234604 21A8 ->  734
27640 : 61344FCDAC6B18DBFE3F -> 1272
27650 : 2005CD9F6D1823FE4020 ->  919
27660 : 1A3E0232A8612AAA6123 ->  749
27670 : 7E23CB7F280125856F30 ->  861
27680 : 0124CDCC6D1805DD7700 ->  924
27690 : DD23CB7E37C02318A82A -> 1101
27700 : AA6123E57E21A86134E1 -> 1232
27710 : FEDDC8FEEDC8FEFDC8FE -> 2327
27720 : CB203C23237EFE36282E ->  885
27730 : E607FE0620284604 0E02 ->  659
27740 : CDAC6B7ECBBFFE5B2810 -> 1405
27750 : 3806CDBA6B2318F1DD77 -> 1200
27760 : 00DD232318E9CD336D21 ->  946
27770 : A861360437C921A86134 ->  929
27780 : 34AFC9FEEBC847040E01 -> 1207
27790 : CDAC6B7ECBBFFE5B2030 -> 1429
27800 : E52AAA61237EE1FEE928 -> 1451
27810 : 0BE5CD336D21A86134E1 -> 1180
27820 : 185ADD360028DD360149 ->  778
27830 : DD360258DD3603292BCB ->  930
27840 : 462803DD340237C93805 ->  705
27850 : CDBA6B1839FE3F2005CD -> 1138
27860 : 9F6D1830FE482027237E ->  898
27870 : CBBF2BFE4C3E48201CDD -> 1182
27880 : 360049DD360158E52AAA ->  932
27890 : 617EFEDD2803DD3401DD -> 1236
27900 : 23DD23E1231805DD7700 ->  920
27910 : DD23CB7E200323188206 ->  815
27920 : 10DD21BD61DD7E00FE49 -> 1230
27930 : 200BDD7E01FE58280EFE -> 1041
27940 : 59280ADD2310EADD21BD -> 1088
27950 : 61AFC937C9DD360028DD -> 1265
27960 : 360149DD360258DD23DD ->  970
27970 : 23DD232AAA617EFEDD28 -> 1241
27980 : 03DD34FF2323CB7EDD36 -> 1205
27990 : 002D2004DD36002BDD23 ->  655
28000 : 7ECB7E2802ED4426006F ->  951
28010 : CDCC6DDD360029DD23C9 -> 1291
28020 : DD21B4612AAA613AA861 -> 1163
28030 : 477E0F0F0F0FCD8F6D7E ->  840
28040 : CD8F6D2310F1C9E60FC6 -> 1393
28050 : 30FE3A3802C607DD7700 ->  963
28060 : DD23C9E52AAA613AA861 -> 1318
28070 : 233D20FC5E235621A861 ->  893
28080 : 3434E1237ECBBFFE3F28 -> 1241
28090 : 0A2BE521A86135160018 ->  679
28100 : 01E5EBCDCC6DE1C9EB7A -> 1766
28110 : F6002814210F27ED5230 ->  760
28120 : 03EB182121E703ED5230 ->  929
28130 : 03EB181D216300ED5230 ->  790
28140 : 03EB181921090 0ED5230 ->  696
28150 : 03EB1815EB181811F0D8 -> 1039
28160 : CD186E1118FCCD186E11 ->  988
28170 : 9CFFCD186E11F6FFCD18 -> 1497
28180 : 6E11FFFF3E2F3C1938FC -> 1139
28190 : ED52DD7700DD23C93E02 -> 1180
28200 : CD011611AE61012000CD ->  754
28210 : 3C20C900000000000000 ->  293
```

Listing 3. »Hexcode« (Schluß)

```
27472 21AE61   LD HL,25006
27475 0620     LD B,32
27477 78       LD A,B
27478 77       LD (HL),A
27479 23       INC HL
27480 10FC     DJNZ 27478
27482 3E01     LD A,1
27484 32A861   LD (25000),A
27487 DD21AE61 LD IX,25006
27491 2AAA61   LD HL,(25002)
27494 E5       PUSH HL
27495 CDCC6D   CALL 28108
27498 E1       POP HL
27499 46       LD B,(HL)
27500 04       INC B
27501 0E01     LD C,1
27503 CDAC6B   CALL 27564
27506 CDD76B   CALL 27607
27509 3810     JR C,27527
27511 213F6B   LD HL,27455
27514 7E       LD A,(HL)
27515 CBBF     RES 7,A
27517 DD7700   LD (IX+0),A
27520 CB7E     BIT 7,(HL)
27522 23       INC HL
27523 DD23     INC IX
27525 28F3     JR Z,27514
27527 CD746D   CALL 28020
27530 CD266E   CALL 28198
27533 2AAA61   LD HL,(25002)
27536 ED4BA861 LD BC,(25000)
27540 0600     LD B,0
27542 09       ADD HL,BC
27543 22AA61   LD (25002),HL
27546 3A085C   LD A,(23560)
27549 FE20     CP 32
27551 C8       RET Z
27552 2AAA61   LD HL,(25002)
27555 ED5BAC61 LD DE,(25004)
27559 ED52     SBC HL,DE
27561 38A5     JR C,27472
27563 C9       RET
27564 21CE61   LD HL,25038
27567 CB7E     BIT 7,(HL)
27569 23       INC HL
27570 28FB     JR Z,27567
27572 10F9     DJNZ 27567
27574 0D       DEC C
27575 20F6     JR NZ,27567
27577 C9       RET
27578 D65A     SUB 90
27580 E5       PUSH HL
27581 21876A   LD HL,27319
27584 CB7E     BIT 7,(HL)
27586 23       INC HL
27587 28FB     JR Z,27584
27589 3D       DEC A
27590 20F8     JR NZ,27584
27592 7E       LD A,(HL)
27593 E67F     AND 127
27595 DD7700   LD (IX+0),A
27598 DD23     INC IX
27600 CB7E     BIT 7,(HL)
27602 23       INC HL
27603 28F3     JR Z,27592
27605 E1       POP HL
27606 C9       RET
27607 DD21BD61 LD IX,25021
27611 7E       LD A,(HL)
27612 E67F     AND 127
27614 C8       RET Z
27615 FE5B     CP 91
27617 3805     JR C,27624
27619 CDBA6B   CALL 27578
27622 1844     JR 27692
27624 FE06     CP 6
27626 3014     JR NC,27648
27628 FE04     CP 4
27630 3043     JR NC,27699
27632 2AAA61   LD HL,(25002)
27635 23       INC HL
27636 46       LD B,(HL)
27637 04       INC B
27638 21A861   LD HL,25000
27641 34       INC (HL)
27642 4F       LD C,A
27643 CDAC6B   CALL 27564
27646 18DB     JR 27611
27648 FE3F     CP 63
27650 2005     JR NZ,27657
27652 CD9F6D   CALL 28063
27655 1823     JR 27692
27657 FE40     CP 64
27659 201A     JR NZ,27687
27661 3E02     LD A,2
27663 32A861   LD (25000),A
27666 2AAA61   LD HL,(25002)
27669 23       INC HL
27670 7E       LD A,(HL)
27671 23       INC HL
27672 CB7F     BIT 7,A
27674 2801     JR Z,27677
27676 25       DEC H
27677 85       ADD A,L
27678 6F       LD L,A
27679 3001     JR NC,27682
27681 24       INC H
27682 CDCC6D   CALL 28108
27685 1805     JR 27692
27687 DD7700   LD (IX+0),A
27690 DD23     INC IX
27692 CB7E     BIT 7,(HL)
27694 37       SCF
27695 C0       RET NZ
27696 23       INC HL
27697 18A8     JR 27611
27699 2AAA61   LD HL,(25002)
27702 23       INC HL
27703 E5       PUSH HL
27704 7E       LD A,(HL)
27705 21A861   LD HL,25000
27708 34       INC (HL)
27709 E1       POP HL
27710 FEDD     CP 221
27712 C8       RET Z
27713 FEED     CP 237
27715 C8       RET Z
27716 FEFD     CP 253
27718 C8       RET Z
27719 FECB     CP 203
27721 203C     JR NZ,27783
27723 23       INC HL
27724 23       INC HL
27725 7E       LD A,(HL)
27726 FE36     CP 54
27728 282E     JR Z,27776
27730 E607     AND 7
27732 FE06     CP 6
27734 2028     JR NZ,27776
27736 46       LD B,(HL)
27737 04       INC B
27738 0E02     LD C,2
27740 CDAC6B   CALL 27564
27743 7E       LD A,(HL)
27744 CBBF     RES 7,A
27746 FE5B     CP 91
27748 2810     JR Z,27766
27750 3806     JR C,27758
27752 CDBA6B   CALL 27578
27755 23       INC HL
27756 18F1     JR 27743
27758 DD7700   LD (IX+0),A
27761 DD23     INC IX
27763 23       INC HL
27764 18E9     JR 27743
27766 CD336D   CALL 27955
27769 21A861   LD HL,25000
27772 3604     LD (HL),4
27774 37       SCF
27775 C9       RET
27776 21A861   LD HL,25000
27779 34       INC (HL)
27780 34       INC (HL)
27781 AF       XOR A
27782 C9       RET
27783 FEEB     CP 235
27785 C8       RET Z
27786 47       LD B,A
27787 04       INC B
27788 0E01     LD C,1
27790 CDAC6B   CALL 27564
27793 7E       LD A,(HL)
27794 CBBF     RES 7,A
27796 FE5B     CP 91
27798 2030     JR NZ,27848
27800 E5       PUSH HL
27801 2AAA61   LD HL,(25002)
27804 23       INC HL
27805 7E       LD A,(HL)
27806 E1       POP HL
27807 FEE9     CP 233
27809 280B     JR Z,27822
27811 E5       PUSH HL
27812 CD336D   CALL 27955
27815 21A861   LD HL,25000
27818 34       INC (HL)
27819 E1       POP HL
27820 185A     JR 27912
27822 DD360028 LD (IX+0),40
27826 DD360149 LD (IX+1),73
27830 DD360258 LD (IX+2),88
27834 DD360329 LD (IX+3),41
27838 2B       DEC HL
27839 CB46     BIT 0,(HL)
27841 2803     JR Z,27846
27843 DD3402   INC (IX+2)
27846 37       SCF
27847 C9       RET
27848 3805     JR C,27855
27850 CDBA6B   CALL 27578
27853 1839     JR 27912
27855 FE3F     CP 63
27857 2005     JR NZ,27864
27859 CD9F6D   CALL 28063
27862 1830     JR 27912
27864 FE48     CP 72
27866 2027     JR NZ,27907
27868 23       INC HL
27869 7E       LD A,(HL)
27870 CBBF     RES 7,A
27872 2B       DEC HL
27873 FE4C     CP 76
27875 3E48     LD A,72
27877 201C     JR NZ,27907
27879 DD360049 LD (IX+0),73
27883 DD360158 LD (IX+1),88
27887 E5       PUSH HL
27888 2AAA61   LD HL,(25002)
27891 7E       LD A,(HL)
27892 FEDD     CP 221
27894 2803     JR Z,27899
27896 DD3401   INC (IX+1)
27899 DD23     INC IX
27901 DD23     INC IX
27903 E1       POP HL
27904 23       INC HL
27905 1805     JR 27912
27907 DD7700   LD (IX+0),A
27910 DD23     INC IX
27912 CB7E     BIT 7,(HL)
27914 2003     JR NZ,27919
27916 23       INC HL
27917 1882     JR 27793
27919 0610     LD B,16
27921 DD21BD61 LD IX,25021
27925 DD7E00   LD A,(IX+0)
27928 FE49     CP 73
27930 200B     JR NZ,27943
27932 DD7E01   LD A,(IX+1)
27935 FE58     CP 88
27937 280E     JR Z,27953
27939 FE59     CP 89
27941 280A     JR Z,27953
27943 DD23     INC IX
27945 10EA     DJNZ 27925
27947 DD21BD61 LD IX,25021
27951 AF       XOR A
27952 C9       RET
27953 37       SCF
27954 C9       RET
27955 DD360028 LD (IX+0),40
27959 DD360149 LD (IX+1),73
27963 DD360258 LD (IX+2),88
27967 DD23     INC IX
27969 DD23     INC IX
27971 DD23     INC IX
27973 2AAA61   LD HL,(25002)
27976 7E       LD A,(HL)
27977 FEDD     CP 221
27979 2803     JR Z,27984
27981 DD34FF   INC (IX-1)
27984 23       INC HL
27985 23       INC HL
27986 CB7E     BIT 7,(HL)
27988 DD36002D LD (IX+0),45
27992 2004     JR NZ,27998
27994 DD36002B LD (IX+0),43
27998 DD23     INC IX
28000 7E       LD A,(HL)
28001 CB7E     BIT 7,(HL)
28003 2802     JR Z,28007
28005 ED44     NEG
28007 2600     LD H,0
28009 6F       LD L,A
28010 CDCC6D   CALL 28108
28013 DD360029 LD (IX+0),41
28017 DD23     INC IX
28019 C9       RET
28020 DD21B461 LD IX,25012
28024 2AAA61   LD HL,(25002)
28027 3AA861   LD A,(25000)
28030 47       LD B,A
28031 7E       LD A,(HL)
28032 0F       RRCA
28033 0F       RRCA
28034 0F       RRCA
28035 0F       RRCA
28036 CD8F6D   CALL 28047
28039 7E       LD A,(HL)
28040 CD8F6D   CALL 28047
28043 23       INC HL
28044 10F1     DJNZ 28031
28046 C9       RET
28047 E60F     AND 15
28049 C630     ADD A,48
28051 FE3A     CP 58
28053 3802     JR C,28057
28055 C607     ADD A,7
28057 DD7700   LD (IX+0),A
28060 DD23     INC IX
28062 C9       RET
28063 E5       PUSH HL
28064 2AAA61   LD HL,(25002)
28067 3AA861   LD A,(25000)
28070 23       INC HL
28071 3D       DEC A
28072 20FC     JR NZ,28070
28074 5E       LD E,(HL)
28075 23       INC HL
28076 56       LD D,(HL)
28077 21A861   LD HL,25000
28080 34       INC (HL)
28081 34       INC (HL)
28082 E1       POP HL
28083 23       INC HL
28084 7E       LD A,(HL)
28085 CBBF     RES 7,A
28087 FE3F     CP 63
28089 280A     JR Z,28101
28091 2B       DEC HL
28092 E5       PUSH HL
28093 21A861   LD HL,25000
28096 35       DEC (HL)
28097 1600     LD D,0
28099 1801     JR 28102
28101 E5       PUSH HL
28102 EB       EX DE,HL
28103 CDCC6D   CALL 28108
28106 E1       POP HL
28107 C9       RET
28108 EB       EX DE,HL
28109 7A       LD A,D
28110 F600     OR 0
28112 2814     JR Z,28134
28114 210F27   LD HL,9999
28117 ED52     SBC HL,DE
28119 3003     JR NC,28124
28121 EB       EX DE,HL
28122 1821     JR 28157
28124 21E703   LD HL,999
28127 ED52     SBC HL,DE
28129 3003     JR NC,28134
28131 EB       EX DE,HL
28132 181D     JR 28163
28134 216300   LD HL,99
28137 ED52     SBC HL,DE
28139 3003     JR NC,28144
28141 EB       EX DE,HL
28142 1819     JR 28169
28144 210900   LD HL,9
28147 ED52     SBC HL,DE
28149 3003     JR NC,28154
28151 EB       EX DE,HL
28152 1815     JR 28175
28154 EB       EX DE,HL
28155 1818     JR 28181
28157 11F0D8   LD DE,55536
28160 CD186E   CALL 28184
28163 1118FC   LD DE,64536
28166 CD186E   CALL 28184
28169 119CFF   LD DE,65436
28172 CD186E   CALL 28184
28175 11F6FF   LD DE,65526
28178 CD186E   CALL 28184
28181 11FFFF   LD DE,65535
28184 3E2F     LD A,47
28186 3C       INC A
28187 19       ADD HL,DE
28188 38FC     JR C,28186
28190 ED52     SBC HL,DE
28192 DD7700   LD (IX+0),A
28195 DD23     INC IX
28197 C9       RET
28198 3E02     LD A,2
28200 CD0116   CALL 5633
28203 11AE61   LD DE,25006
28206 012000   LD BC,32
28209 CD3C20   CALL 8252
28212 C9       RET
```

Listing 4. »Dissassembler-Listing«

3D-Grafik

Es gibt gute 3D-Grafikprogramme, die sind aber meist sehr lang. Dieses hier macht einige Kompromisse in der Bedienung, ist dafür aber extrem kurz.

Das folgende Programm besticht durch seine Kürze und Einfachheit und ist dadurch leicht einzutippen. Es stellt erstaunlich dreidimensionale Grafiken dar. Verdeckte Linien werden erkannt und weggelassen, das heißt, daß Flächen oder Linien, die man bei einem undurchsichtigem Körper nicht sehen würde, auch nicht geplottet werden.

Prinzipiell kann man zwar jede Funktion verwenden, ich habe mich aber auf vier recht interessante Kurven beschränkt. Wie man das Programm beliebig erweitern kann, erkläre ich später.

Wie entsteht eine Grafik?

Ein beliebiger Abschnitt der zu plottenden Funktion wird um eine Achse gedreht, die am linken Ende des Grafen parallel zur Ordinate steht. Von oben gesehen gibt diese Drehung eine kreisförmige Fläche.

Würde man dieses Gebilde in der Mitte durchschneiden und von der Seite aus betrachten, so sähe man den Funktionsausschnitt in doppelter Ausführung — links die Spiegelung und rechts das Original. (siehe Skizze!)

Mein Programm berechnet eine Draufsicht von vorne auf die sich ergebende Rotationsfigur.

Wie benutze ich das Programm?

Nachdem Sie das Listing eingetippt haben, starten Sie es durch »RUN«.

Jetzt können Sie die Form der gewünschten Grafik wählen. Dabei stehen Ihnen, wie schon erwähnt, vier Grundversionen zur Verfügung. Nun geben Sie den Verzerrungsfaktor ein, der zwischen 0 und 3 liegen muß. Null bedeutet gerade Draufsicht von vorne. Je größer dieser Faktor ist, desto seitlicher sehen Sie auf die Fläche.

Tja — und jetzt brauchen Sie nur noch etwas Geduld. Die Prozedur dauert nämlich oft über eine halbe Stunde, was nicht an den benötigten Berechnungen liegt, sondern an Ihrem Spectrum, der sich für die hunderte zu plottenden Pixel reichlich Zeit läßt.

Dafür werden Sie aber am Ende durch faszinierende Ansichten belohnt, die natürlich mit »COPY« auf Ihrem Drucker verewigt werden können.

Falls Ihnen das Programm nach den ersten Minuten doch zu langsam erscheint, so können Sie es beschleunigen, indem Sie in der Zeile 190 die Schrittweite erhöhen:
zum Beispiel 190 FOR x=—x0 TO x0 STEP 4*10

Durch die jetzt erreichte Geschwindigkeit werden die Kurven gröber, da Zwischenwerte von Geraden verbunden werden und damit viele Berechnungen erspart bleiben.

Mathematisch begabten Tüftlern wird es leicht fallen, die Zeilen 340—370 für andere Funktionen ihrer Wahl abzuändern.

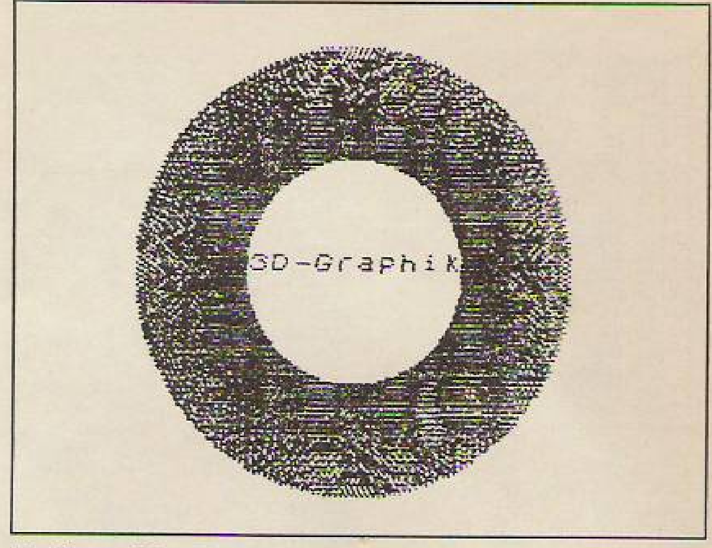
Das Grundprinzip ist Folgendes:
340 LET yy=»Funktion f(xa)« * Z

Den Faktor Z am Ende der Zeile müssen Sie experimentell bestimmen.

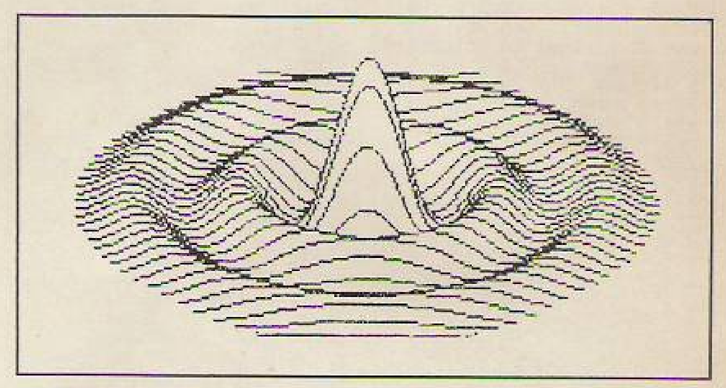
Falls Sie das Bild durchsichtig, also mit verdeckte Linien wünschen, so löschen Sie die Zeile 150.

Probieren Sie ruhig einmal ein paar Veränderungen, es wird bestimmt Spaß machen.

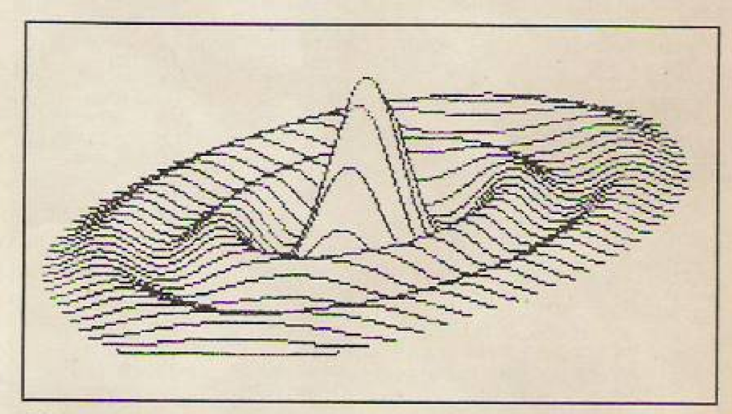
(Martin Hilpert)



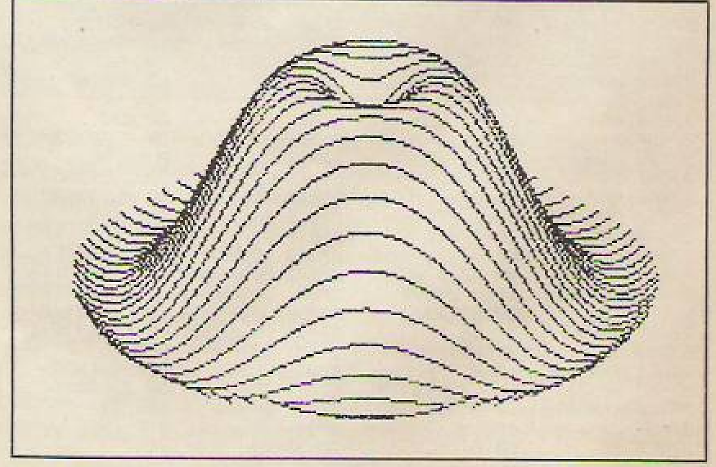
Hardcopy (Menü)



SCREEN$ 1, Zerrfaktor 0



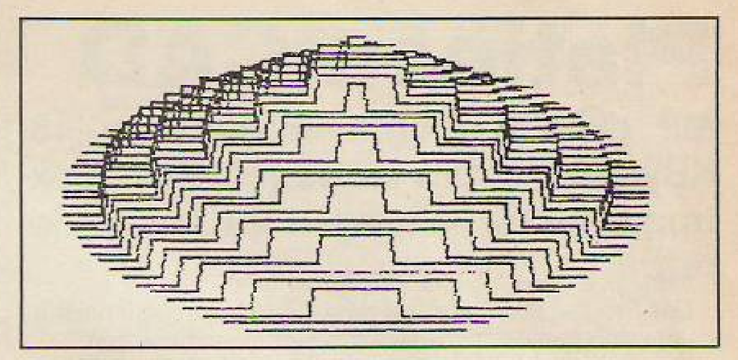
SCREEN$ 1, Zerrfaktor 3



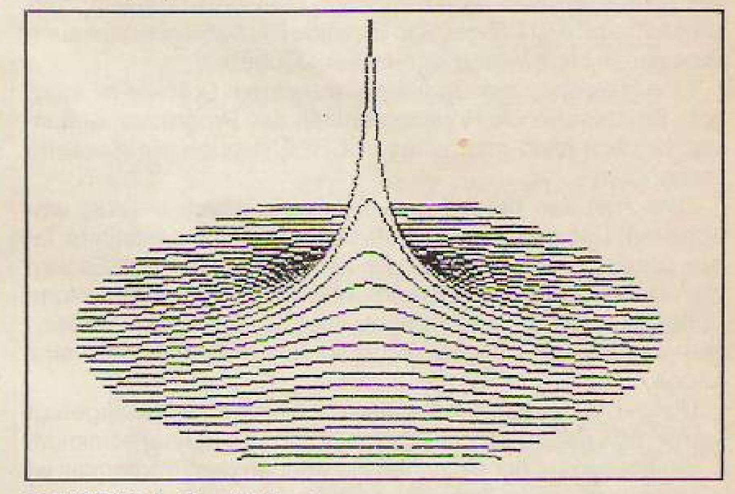
SCREEN$ 2, Zerrfaktor 0

```
  10 REM
     ###########################
     #                         #
     #      "3D-Graphic"       #
     #                         #
     # © by Martin W. Hilpert  #
     #      Kopernikusweg 4    #
     #       8630 Coburg       #
     #    (Tel.09561/30173)    #
     #        Jan. 1985        #
     #                         #
     ###########################
  20 BEEP 1,20: BORDER 7: PAPER
7: INK 0: CLS
  30 PLOT 50,65: DRAW 100,100,21
09: BEEP 1,40
  40 DIM z(256): LET xx=0
 100 PRINT OVER 1;AT 10,11;"3D-G
raphik"
 110 LET a$="Gib den Typ der Gra
phik ein!    (1/2/3/4)  ": INPUT
(a$);typ: IF typ<0 OR typ>4 THEN
 GO TO 110
 120 LET a$="Gib den Verzerrungs
faktor ein!  0,1,2,3   (normaler
weise:0)  ": INPUT (a$);ve: CLS
 130 IF ve>3 OR ve<0 THEN GO TO
120
 140 REM Anzahl der Kurven:
 145:
 150 FOR z=-17 TO +17
 160 LET x0=430-(z*z)+1.3*ABS z
 170 LET zz=z*11: LET xx=xx+ve
 175:
 180 REM Anzahl der Punkte:
 185:
 190 FOR x=-x0 TO x0 STEP 4*1
 200 GO SUB 300
 210 NEXT x: NEXT z
 220 FOR i=10 TO 55: BEEP .001,i
: NEXT i: STOP
 225:
 300 REM Berechnung des
         Argumentes:
 305:
 310 LET xa=SQR (x*x+5*zz*zz)/25
 315:
 320 REM Berechnung des
         Funktionswertes
 325:
 330 GO TO 330+typ*10
 340 LET yy=(SIN xa/xa)*50: GO T
O 390
 350 LET yy=SIN (xa/3.5+.4)*40+8
: GO TO 390
 360 LET yy=50-INT (xa/3)*10: GO
 TO 390
 370 LET yy=45-(LN (xa*2)-xa/20)
*25
 380 REM Berechnung der
         Koordinaten:
 385:
 390 LET x1=INT (x/4)+127+xx-20*
ve
 400 LET y1=INT (yy+zz/4)+80
 405:
 410 REM Abfrage der verdeckten
         Linien:
 415:
 420 IF y1<=z(x1) THEN GO TO 480
 425:
 430 REM Plotten der Funktion:
 435:
 440 PLOT x1,y1
 450 LET z(x1)=y1
 455:
 460 REM Liegen die Punkte nicht
         direkt nebeneinander
         -> ziehe Linie:
 465:
 470 IF x0>x AND x>-x0 THEN DRAW
x2-x1,y2-y1
 480 LET x2=x1: LET y2=y1
 490 RETURN
 500 REM          ENDE
```

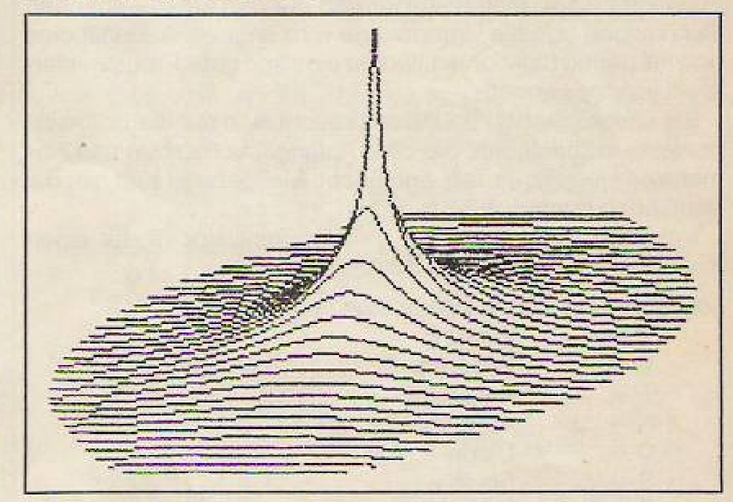
Basic-Listing »3D-Grafik«



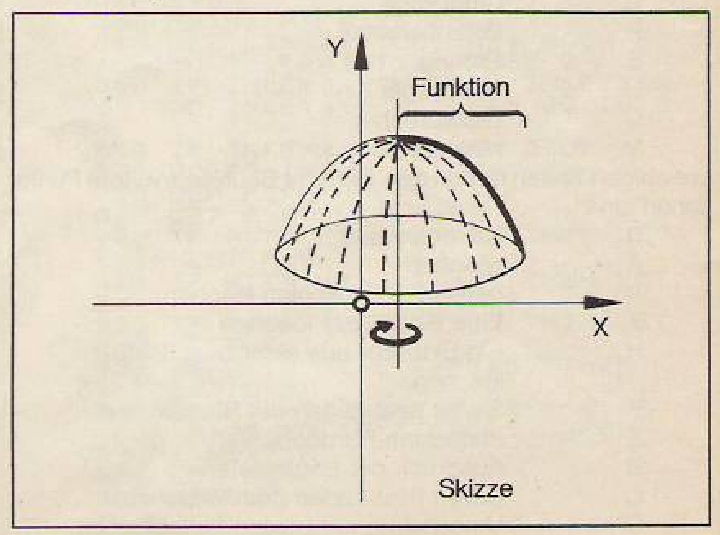
SCREEN$ 3, Zerrfaktor 0



SCREEN$ 4, Zerrfaktor 0



SCREEN$ 4, Zerrfaktor 3



Funktions-Skizze

Platinen-CAD

Mit diesem Programm für den 48-KByte-Spectrum ist es möglich, elektronische Schaltungen zu konstruieren.

Das Programm arbeitet mit zwei Grafikseiten, das heißt es lassen sich zwei Bildschirme mit Schaltungen besetzen.

Es wird eine automatische Bauteilliste geführt: Zeichnet man ein Bauteil, so wird es automatisch numeriert und sein Wert wird mittels INPUT abgefragt. Einzelne Bauteilwerte oder auch eine ganze Liste lassen sich später löschen.

Eine Hardcopy der Grafikseite auf einem Drucker ist möglich. Entsprechende Routinen enthält das Programm. Selbstverständlich kann man einen »SCREEN« auch auf Kassette aufnehmen.

Zwischen den beiden Bildschirmen läßt sich beliebig umschalten. Das geschieht mittels Druck auf eine bestimmte Taste, dazu später noch mehr. Hier sei nur soviel gesagt: Es wird zum Umschalten eine Assembler-Routine verwendet, welche außerordentlich schnell arbeitet. Auch die Bauteillisten lassen sich ausdrucken oder auf Band speichern beziehungsweise Laden.

Übrigens: Zu dem Programm darf nichts mehr hinzugefügt werden, da es den ganzen Speicherplatz (48 KByte) einnimmt.

Die Steuerung der einzelnen Funktionen des Programms ist sehr einfach. Jede Taste wurde mit einem Bauteil oder einer Funktion belegt. Man bestätigt also nur den entsprechenden Buchstaben und die Unterroutine wird angewählt. Damit dies schnell genug funktioniert, wurde eine spezielle Programmiertechnik angewendet.

Bei elektronischen Bauteilen braucht man nur die Buchstabentaste zu betätigen, die dem Anfangsbuchstaben des Namens des jeweiligen Teils entspricht. Maßgebend sind hier die deutschen Namen.

Für Widerstand tippt man »W«, für Transistor »T«, für einen Kondensator »K« und so weiter.

Folgende Funktionen lassen sich ausführen:

Taste	Wirkung
W	Widerstand
T	Transistor
K	Kondensator
D	Diode
S	Spule
B	Batterie
G	Glühlampe
P	Potentiometer
E	Erdung
U	Übertrager
L	Lautsprecher
M	Mikrofon

Die obigen Tasten führen also allesamt Bauteile. Weitere Funktionen sind:

O	Lötverbindung
A	Löschen
V	gesamten Bildschirm löschen
Y	Eine Bauteilliste löschen
H	Ein Einzelteil aus einer best. Liste löschen
F	Saven bzw. Laden der Bauteillisten
Z	Bildschirm-Hardcopy
R	Ausdruck der Bauteillisten
Q	Saven bzw. Laden des Bildschirms
C	Umschalten der beiden Bildschirme

Als Cursor dient ein PLOT-Punkt (Pixel), der sich in einem blinkendem Printfeld befindet. Bewegt wird dieser Punkt mit den Tasten 5-8. Mit der Taste »0« wird ein Pixel gesetzt, und dieses ist dann der Anfangspunkt einer zu zeichnenden Linie.

Also, um eine Linie zu zeichnen betätigt man die Taste »0« an der gewünschten Bildschirmstelle. Die Bildschirmumrandung ist dann rot. Nun fährt man mit dem Cursor zu dem gewünschten Endpunkt der Linie und betätigt nochmals die Taste »0«. Die Bildschirmumrandung ist dann grün. Man muß also immer Anfangs- und Endpunkt der Linie lediglich mit »0« markieren; das Zeichnen der Linie übernimmt der Computer selber.

Noch etwas zum Cursor. Betätigt man die Tasten 5-8 zusätzlich zusammen mit SHIFT, so läßt sich der Cursor um jeweils eine PLOT Position verschieben. Somit ist also ein genaues Arbeiten gewährleistet.

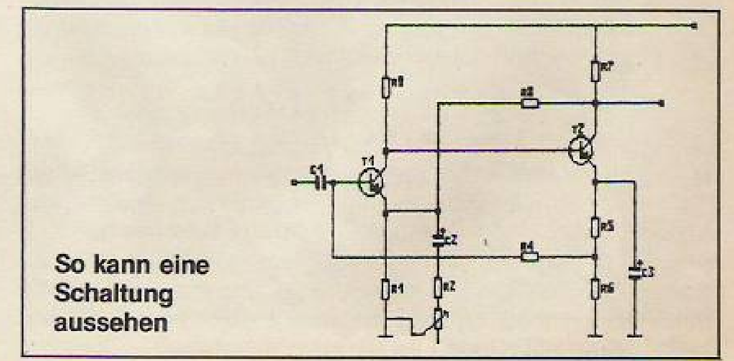
Nun etwas zum eigentlichen Zeichnen der elektronischen Bauteile: Man bewegt den Cursor an die gewünschte Stelle irgendwo auf dem Bildschirm und betätigt dann eine beliebige Bauteiltaste, zum Beispiel »W«. Nun fragt das Programm unten am Bildschirm die Richtung ab, in die das Bauteil gezeichnet werden soll. Die möglichen Richtungen bei allen Bauteilen sind: Norden, Süden, Westen und Osten. Diese vier Richtungen werden wieder mit den Cursortasten 5-8 bestimmt. Möchte man zum Beispiel ein Bauteil in Richtung Westen (immer relativ zum Cursor) gezeichnet haben, so betätigt man Taste »5« und so weiter. Die Bedienung ist somit einfach gehalten.

Hat man einmal das falsche Unterprogramm angewählt, so kann man es durch betätigen der Taste »0« wieder mühelos verlassen. Dies gilt in gleichem Maße für alle Funktionen, außer für die, die andere Antworten, wie ja oder nein verlangen.

Ist das Bauteil gezeichnet, so muß noch sein Wert für die automatische Liste eingegeben werden. Dies trifft jedoch nur für Widerstände, Transistoren, Dioden, Kondensatoren, Potentiometer und Spulen zu. Bei allen anderen Bauteilen muß kein Wert eingetippt werden. Obengenannte Bauteile haben jeweils eine eigene Bauteilliste für sich alleine. Alle anderen Bauteile (wie Batterie, Lautsprecher, Mikrofon, Erdung und so weiter) haben eine gemeinsame Liste. Widerstände, Transistoren, Dioden, Kondensatoren und Spulen werden automatisch mit fortlaufenden Ziffern numeriert. Insgesamt lassen sich 240 Bauteile speichern. Pro Bauteilgruppe 40 Stück. Ist diese Zahl erreicht, sollte man sich die Bauteilliste ausdrucken lassen, um



Die zur Verfügung stehenden Schaltzeichen des Programms



So kann eine Schaltung aussehen

weitermachen zu können, ohne Bauteile zu löschen. Im allgemeinen wird man aber nicht soviele Bauteile benötigen. Noch etwas ist dazu zu sagen: Bei der Eingabe der Werte können insgesamt 8 beliebige Zeichen eingetippt werden. Außer dem reinen Zahlenwert lassen sich also auch noch andere Kurzinformationen eingeben. Bei der Liste, die die diversen Bauteile (Batterie, Lautsprecher, Erdung usw.) führt, wird der Name des verwendeten Bauteils automatisch gespeichert.

Zur Verdeutlichung der hier gemachten schriftlichen Erläuterungen, sollte man auch die beigelegten Hardcopys heranziehen, die mit dem Programm angefertigt wurden. Übrigens wurden sie auf einem Seikosha GP-50 S erstellt, der sich hierfür bestens eignet.

Für die auf der Hardcopy wiedergegebene Schaltung wurden etwa 5 Minuten benötigt, um sie aufzubauen. Die Bedienung des Programm ist wirklich einfach. Um es noch einmal deutlich zu sagen:

Für ein bestimmtes Bauteil, tippt man den Anfangsbuchstaben des Namens des Bauteils an, also: »W« für Widerstand, »B« für Batterie zum Beispiel.

Und jetzt erfolgen detaillierte Erklärungen der einzelnen Möglichkeiten und Funktionen des Programms. Man kann diese Liste neben den Computer legen, wenn man mit diesem Programm arbeitet.

Taste »W«
Zeichnet in vier Himmelsrichtungen einen **Widerstand**. Automatische Numerierung und Listenführung. Wert muß eingetippt werden.

Taste »T«
Zeichnet in vier Himmelsrichtungen einen **Transistor**. Es kann zwischen PNP oder NPN gewählt werden. Automatische Numerierung und Listenführung. Bezeichnung (zum Beispiel BC 238) muß eingetippt werden.

Taste »K«
Zeichnet in vier Himmelsrichtungen einen **Kondensator**. Es kann zwischen einem ungepolten und einem gepolten gewählt werden (geplot=Elko). Automatische Numerierung und Listenführung. Wert muße eingetippt werden.

Taste »D«
Zeichnet in vier Himmelsrichtungen eine **DIODE**. Es kann zwischen normaler Diode und Zenerdiode gewählt werden. Automatische Listenführung und Numerierung. Wert muß eingetippt werden.

Taste »S«
Zeichnet in vier Himmelsrichtungen eine **Spule**. Automatische Numerierung und Listenführung. Wert muß eingetippt werden.

Taste »P«
Zeichnet in vier Himmelsrichtungen ein **Potentiometer**. Automatische Listenführung. Wert muß eingetippt werden (Zählt zu den Widerständen; Unterscheidung macht man am besten durch Zusatz von P.).

Taste »B«
Zeichnet in vier Himmelsrichtungen eine **Batterie**. Automatische Listenführung.

Taste »G«
Zeichnet in vier Himmelsrichtungen **Glühlampe**. Automatische Listenführung.

Taste »E«
Zeichnet in vier Himmelsrichtungen das **Erdungssymbol**. Automatische Listenführung.

Taste »U«
Zeichnet in vier Himmelsrichtungen einen **Übertrager (Transformator)**. Automatische Listenführung.

Taste »L«
Zeichnet in vier Himmelsrichtungen einen **Lautsprecher**. Automatische Listenführung.

Taste »M«
Zeichnet in vier Himmelsrichtungen ein **Mikrofon**. Automatische Listenführung.

Taste »O« (O wie in Vogel!)
Zeichnet an der momentanen Position eine **Lötverbindung**.

Taste »A«
Es wird der Bereich gelöscht, der blinkt (eine Printposition). Muß man ausprobieren.

Die oben aufgezeigten Funktionen haben alle etwas mit dem Editieren einer Schaltung zu tun. Gemeinsam haben alle obengenannten Funktionen, daß sie immer von der momentanen Position des Cursors (PLOT-Punkt, blinkend) ausgehen.

Weitere Funktionen

Taste »V«
Bildschirm löschen
Man kann den (momentan sichtbaren) Bildschirm löschen. Tippt man diese Taste wird aber nicht sofort gelöscht, sondern erst noch einmal gegengefragt.

Taste »Y«
Mit dieser Funktion kann man eine Bauteilliste löschen. Gelöscht wird vollständig und erst nach ausdrücklichem Wunsch. Auch hier kann man mit Taste »0« verneinen, jedoch nur, bevor man den Buchstaben des Namens der Liste noch nicht eingetippt hat (R,T,D,C,L, oder E).

Taste »H«
Hiermit kann man ein spezielles Bauteil aus einer bestimmten Liste löschen. Man gibt zuerst die Nummer des Bauteils ein und dann die Bauteilart.

Taste »F«
Mit dieser Funktion kann man die Bauteillisten Laden oder auf Band sichern. Man verwendet am besten eine spezielle Kassette. In der Funktion »SAVE« wird ein VERIFY-Durchgang gemacht. Dafür muß das Band wieder zurückgespult werden. Tritt ein Tape Loading Error auf, so gibt man GOTO 4000 ein und wiederholt den SAVE-Vorgang. GOTO 4000 kann immer angewendet werden, wenn man aus irgendeinem Grund aus dem Programm gekommen ist. Daten gehen hierbei nicht verloren.

Das SAVE der Bauteillisten kann sehr vorteilhaft sein, wenn man die Konstruierung des Schaltplans zu einem späteren Zeitpunkt fortsetzen möchte, zum Beispiel am anderen Tag. Dazu sichert man beide Grafikseiten auf Band (siehe Taste »Q«) und die Bauteilliste. Später lädt man dann zuerst das Hauptprogramm und dann die Bauteillisten und die beiden Bildschirme.
Mit der Taste »F« wird die Bauteilliste auch geladen.

Taste »Z«
Hiermit kann man eine Hardcopy des momentan sichtbaren Bildes auf dem angeschlossenen Drucker ausgeben lassen. Möchte man beide Bildschirme ausdrucken lassen, so läßt man erstmal den einen ausdrucken, schaltet dann auf den zweiten um (siehe Taste »C«) und läßt den ebenfalls wieder mit Hilfe der Taste »Z« ausdrucken. Als Drucker eignen sich der kleine Sinclair, der Alphacom und der Seikosha GP-50 sowie der GP-100 gut. Vorrang gebe ich jedoch den beiden Seikosha-Druckern, da sie im allgemeinen ein deutlicheres Druckbild abgeben, das sich auch mühelos kopieren läßt!

Taste »R«
Hiermit lassen sich die Bauteillisten ausdrucken.

Taste »Q«
Mit dieser Funktion läßt sich der momentane Bildschirm abspeichern, beziehungsweise ein SCREEN von Band laden.

Taste »C«
Diese Funktion ist sehr wichtig. Sie gestattet es zwischen zwei Grafikseiten blitzschnell umzuschalten. Jeweils eine Grafikseite ist auf dem Bildschirm zu sehen, und die andere ist über RAMTOP gespeichert und ist somit auch vor NEW geschützt. Jedesmal wenn man die Taste »C« tippt, wird zwischen den beiden Grafikseiten umgeschaltet, das heißt sie werden aus-

getauscht: Die eine wird gespeichert, die andere wird auf den Bildschirm geholt.

Dazu wird eine ausgeklügelte Maschinencoderoutine verwendet, die diesen Austausch schnell und sicher gewährleistet. Die Routine ist 33 Byte lang und ist ebenfalls über RAMTOP gespeichert.

Noch einmal etwas allgemeines, für alle Funktionen gültiges: Hat man aus Versehen eine nicht gewünschte Routine angewählt, so kann man diese durch die Taste »0« (Null!) wieder verlassen. Hat man das Programm einmal ausprobiert, merkt man erst, wie leicht die Bedienung ist. Am besten, man macht sich eine Stichwortliste mit den wichtigsten Funktionen, die man sich neben den Computer legt.

Da ich das Programm entwickelt habe, um selber elektronische Layouts aufzubauen, gebe ich hier noch einige nützliche Tips zum Aufbauen einer Schaltung.

Am besten ist, man fängt mit den Transistoren an, denn sie bestimmen die Dimensionen der gesamten Schaltung. Dann sollte man Dioden, Widerstände und andere Bauteile einfügen.

Die Verdrahtung und Lötstellen (Taste »O«) kann man zuletzt machen oder fortlaufend an die gezeichneten Teile ansetzen. Ich schlage vor, zuletzt zu verdrahten.

Man kann auch größere Schaltungen zeichnen, als solche über zwei Bildschirme, indem man jeweils einen Bildschirm zeichnet, diesen dann ausdruckt, den Bildschirm löscht und weiter macht.

Um es noch einmal in Erinnerung zu rufen. Insgesamt lassen sich 240 Bauteile speichern, das dürfte selbst für größere Schaltpläne reichen. Und auch wenn die Bauteillisten voll sind, kann man ja die Werte auch »zu Fuß« aufschreiben.

Löscht man ein bereits gezeichnetes Bauteil, so sollte man auch dessen Wert aus seiner Liste löschen. Dazu wählt man mit Taste »H« die Funktion Einzelteil löschen. Man gibt zuerst die Nummer des Bauteils ein, die es auf dem Bildschirm hatte, und dann die Bauteilart, also zum Beispiel R,T,C,D,L oder E. »E« steht für diverse Bauelemente wie Mikrofon, Erdungssymbol, Übertrager und so weiter.

Alle Einzelbuchstaben (außer N) werden für die einzelnen Unterprogramme gebraucht. Sie enthalten Zeilennummern.

xx	horizontale Komponente der Position des Cursors
yy	vertikale Komponente der Position des Cursors
r$()	List mit Zählvariable rr der Widerstände
rr	
c$	Liste mit Zählvariable cc der Kondensatoren
cc	
d$()	Liste mit Zählvariable dd der Dioden
dd	
t$()	Liste mit Zählvariable tt der Transistoren
tt	
l$()	Liste mit Zählvariable ll der Spulen
ll	
e$()	Liste mit Zählvariable ee der diversen Bauteile
ee	
x()	werden gebraucht für Verbindungslinien
y()	
horiz	enthalten wieder bestimmte Zeilennummern; stehen am Anfang eines Unterprogrammes.
Richtung	=9800, ist eine Zeilennummer eines Unterprogramms

Wichtige Variablen

Sollte man, aus welchem Grund auch immer, einmal aus dem Programmablauf gelangen (zum Beispiel bei Tape Loading Error!), so gibt man »GOTO 4000« ein. Daten gehen hierbei nicht verloren, falls man dies sofort ausführt ohne zwischendurch eine andere Taste zu betätigen.

Die Variablenliste ist im allgemeinen nicht so wichtig für das Verständnis des Programms. Die wichtigsten Variablen sind xx und yy. Sie repräsentieren die momentane Position des Cursors.

(M. Malik)

Die meisten REM-Zeilen mußten aus Speicherplatzgründen weggelassen werden. An den entsprechenden Stellen befinden sich lediglich Doppelpunkte (:), damit das Programm nicht zu unübersichtlich wird.

Der Programmaufbau ist im großen und ganzen einfach. Es gibt einen Hauptteil und Unterprogramme.

1000-3000	Variablendefinition. Dies ist ein sehr wichtiger Teil. Es ist beim Eintippen darauf zu achten, daß die Zahlen richtig eingetippt werden, sonst funktioniert das Programm nicht optimal.
4000-4200	Dieser Programmteil steuert den Cursor und die Unterprogramme.
4200-5000	Dieser Teil ist verantwortlich für das Ziehen der Verbindungslinien.
ab 5000	Ab hier folgen nur noch Unterprogramme für die einzelnen Bauteile und Funkionen.
9800	Ab 9800 befindet sich ebenfalls noch ein sehr wichtiges Unterprogramm, das die Richtung abfragt und an die einzelnen Unterprogramme zum Zeichnen der Bauteile weitergibt.

Programmaufbau

```
  10 REM ************************
  20:
  25 REM    PLATINEN-CAD
  30 REM  EIN PROGRAMM VON
  32:
  35 REM  MARKUS MALIK
  40 REM  SCHLESISCHER WEG 4
  45 REM  4775 LIPPETAL
  50:
  55:
  60 REM  FUER ZX-SPECTRUM 48-K
  65 REM      UND DRUCKER
  70:
  75 REM   © MALIK SOFTWARE
  80 REM     JANUAR 1985
  90:
  95:
 100 REM ************************
1000 REM VARIABLEN
1005 CLEAR 53340
1010 GO SUB 8300
1025 LET EI=0
1030 POKE 23300,192: POKE 23301,
23
1110 LET W=5000
1120 LET T=6460
1130 LET C=6400
1140 LET D=5800
1150 LET A=6650
1160 LET S=6670
1170 LET B=6800
1175 LET G=7000
1180 LET O=7200
1190 LET P=6100
1195 LET E=6300
1200 LET U=6390
1210 LET L=7250
1220 LET M=7410
1230 LET I=9990
1240 LET V=7670
1250 LET Y=7700
1260 LET H=7780
1270 LET F=7850
1280 LET Z=7910
1300 LET Q=8050
1310 LET J=9990
1320 LET K=5300
1290 LET R=7950
2000 LET XX=132: LET YY=92
2010 POKE 23562,1: POKE 23658,8
2020 LET RICHTUNG=9800
2025 LET Y$="
"
2030 DIM R$(41,8): LET RR=1
2035 DIM C$(41,8): LET CC=1
2040 DIM D$(41,8): LET DD=1
2045 DIM T$(41,8): LET TT=1
2050 DIM L$(41,8): LET LL=1
2055 DIM E$(41,12): LET EE=1
3000 GO TO 4200: REM ANFANG
```

Basic-Listing »Platinen-CAD«

```
3001:
3002:
4050 LET A$=INKEY$
4053 IF A$="" THEN GO TO 4050
4060 IF A$="0" THEN PLOT OVER 1;
 INK 8; PAPER 8;XX,YY: RETURN
4065 IF A$>="A" AND A$<="Z" AND
A$<>"N" AND A$<>"X" THEN GO SUB
VAL A$: PLOT PAPER 8, INK 8, OVE
R 1, FLASH 1,XX,YY
4070 PLOT PAPER 8; INK 8; OVER 1
; FLASH 0;XX,YY
4080 LET XX=XX+((CODE A$=9)-(COD
E A$=8))+8*((A$="8")-(A$="5"))
4090 LET YY=YY+((CODE A$=11)-(CO
DE A$=10))+8*((A$="7")-(A$="6"))
4105 IF XX>248 THEN LET XX=248:
GO TO 4120
4107 IF XX<0 THEN LET XX=0: GO T
O 4120
4110 IF YY>168 THEN LET YY=168:
GO TO 4120
4115 IF YY<0 THEN LET YY=0
4125 PLOT PAPER 8, INK 8; OVER 1
; FLASH 1;XX,YY
4130 GO TO 4000
4150:
4160:
4200 REM LINIEN
4210 DIM X(2)
4220 DIM Y(2)
4230 FOR N=1 TO 2
4240 GO SUB 4120
4245 PLOT XX,YY
4250 LET X(N)=XX
4270 LET Y(N)=YY
4280 IF INKEY$<>"" THEN GO TO 42
80
4290 BORDER 2
4300 NEXT N
4310 PLOT X(2),Y(2)
4320 DRAW X(1)-X(2),Y(1)-Y(2)
4330 BORDER 4
4340 GO TO 4230
5000:
5002:
5005 LET B$="WIDERSTAND"
5010 LET HORIZ=5120
5015 LET VERTI=5050
5020 GO TO RICHTUNG
5050 IF YY>159 AND AA<>-1 THEN G
O TO 9990
5060 IF YY<16 AND AA<>1 THEN GO
TO 9990
5065 IF XX<2 THEN GO TO 9990
5070 PLOT XX,YY: DRAW 0,AA*3
5075 PLOT XX+2,YY+AA*3: DRAW 0,A
A*10
5080 DRAW -4,0: DRAW 0,-AA*10
5085 DRAW 4,0
5090 PLOT XX,YY+AA*13: DRAW 0,AA
*3
5091 LET BB1=4: LET BB2=8: LET B
B3=12: IF XX>240 THEN LET BB1=-1
4: LET BB2=-10: LET BB3=-6
5092 PLOT XX+BB1,YY+AA*9: DRAW 0
,3: DRAW 2,0: DRAW 0,-1: DRAW -1
,-1: DRAW 1,-1
5093 LET M$=STR$ RR: LET XXX=XX+
BB2: LET YYY=YY+AA*9: GO SUB 810
0+10*VAL M$(1)
5094 IF LEN M$=2 THEN LET XXX=XX
+BB3: GO SUB 8100+10*VAL M$(2)
5100 GO TO 5190
5120 IF XX<16 AND AA<>1 THEN GO
TO 9990
5130 IF XX>232 AND AA<>-1 THEN G
O TO 9990
5135 IF YY<2 THEN GO TO 9990
5140 PLOT XX,YY: DRAW AA*3,0
5150 PLOT XX+AA*3,YY+2: DRAW AA*
10,0
5160 DRAW 0,-4: DRAW -AA*10,0
5170 DRAW 0,4
5180 PLOT XX+AA*13,YY: DRAW AA*3
,0
5181 LET BB1=3: LET BB2=7: LET B
B3=11: IF AA=-1 THEN LET BB1=-13
: LET BB2=-9: LET BB3=-5
5182 PLOT XX+BB1,YY+4: DRAW 0,3:
 DRAW 2,0: DRAW 0,-1: DRAW -1,-1
: DRAW 1,-1
5183 LET M$=STR$ RR: LET XXX=XX+
BB2: LET YYY=YY+4: IF YY=168 THE
N LET YYY=YYY-1
5184 GO SUB 8100+10*VAL M$(1): L
ET XXX=XX+BB3: IF LEN M$<>2 THEN
 GO TO 5190
5186 GO SUB 8100+10*VAL M$(2)
5190 INPUT "WERT: "; LINE R$(RR)
5200 LET RR=RR+1: IF RR=41 THEN
LET RR=1
5210 RETURN
5300:
5303:
5310 LET B$="KONDENSATOR"
5320 LET HORIZ=5370
5330 LET VERTI=5450
5340 GO TO RICHTUNG
5370 LET RICH=0
5380 IF XX>232 AND AA<>-1 THEN G
O TO 9990
5390 IF XX<16 AND AA<>1 THEN GO
TO 9990
5395 IF YY<3 OR YY>166 THEN GO T
O 9990
5400 PLOT XX,YY: DRAW AA*6,0
5405 PLOT XX+AA*5,YY-3: DRAW 0,6
5410 PLOT XX+AA*6,YY-3: DRAW 0,6
5415 PLOT XX+AA*9,YY-3: DRAW 0,6
5420 PLOT XX+AA*10,YY-3: DRAW 0,
6
5425 PLOT XX+AA*11,YY: DRAW AA*5
,0
5426 LET BB1=5: LET BB2=7: LET B
B3=11: IF AA=-1 THEN LET BB1=-11
: LET BB2=-9: LET BB3=-5
5427 PLOT XX+BB1,YY+5: DRAW -2,0
: DRAW 0,3: DRAW 2,0
5428 LET XXX=XX+BB2: LET YYY=YY+
5: LET M$=STR$ CC: GO SUB 8100+1
0*VAL M$(1)
5429 LET XXX=XX+BB3: IF LEN M$=2
 THEN GO SUB 8100+10*VAL M$(2)
5435 GO TO 5510
5450 LET RICH=1
5460 IF YY>159 AND AA<>-1 THEN G
O TO 9990
5470 IF YY<16 AND AA<>1 THEN GO
TO 9990
5475 IF XX<3 THEN GO TO 9990
5480 PLOT XX,YY: DRAW 0,AA*5
5485 PLOT XX-3,YY+AA*5: DRAW 6,0
5490 PLOT XX-3,YY+AA*6: DRAW 6,0
5495 PLOT XX-3,YY+AA*9: DRAW 6,0
5500 PLOT XX-3,YY+AA*10: DRAW 6,
0
5501 PLOT XX,YY+AA*11: DRAW 0,AA
*5
5502 LET BB1=7: LET BB2=9: LET B
B3=13: IF XX>240 THEN LET BB1=-1
3: LET BB2=-11: LET BB3=-7
5503 PLOT XX+BB1,YY+AA*9: DRAW -
2,0: DRAW 0,3: DRAW 2,0
5504 LET XXX=XX+BB2: LET YYY=YY+
AA*9: LET M$=STR$ CC: GO SUB 810
0+10*VAL M$(1)
5505 IF CC>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
5520 PRINT #0;AT 1,0;" GEPOLT ?
--    (JA=5,NEIN=8)       "
5530 IF INKEY$<>"" THEN GO TO 55
30
5540 IF INKEY$="8" THEN GO TO 57
20
5550 IF INKEY$="5" THEN GO TO 55
80
5560 GO TO 5540
5580 PRINT #0;AT 1,0;"PLUSPOL ?(
RICHTUNG MIT CURSOR !)"
5590 IF RICH=1 THEN GO TO 5660
5600 IF INKEY$<>"" THEN GO TO 56
00
5610 IF INKEY$="8" THEN LET BB=1
: GO TO 5640
5620 IF INKEY$="5" THEN LET BB=-
1: GO TO 5640
5630 GO TO 5610
```

```
5640 PLOT XX+AA*8+BB*6,YY+2: DRA
W 0,2
5645 PLOT XX+AA*8+BB*6-1,YY+3: D
RAW 2,0
5650 GO TO 5720
5660 IF INKEY$<>"" THEN GO TO 56
60
5670 IF INKEY$="6" THEN LET BB=-
1: GO TO 5700
5680 IF INKEY$="7" THEN LET BB=1
: GO TO 5700
5690 GO TO 5670
5700 PLOT XX+2,YY+AA*8+BB*6: DRA
W 2,0
5705 PLOT XX+3,YY+AA*8+BB*6-1: D
RAW 0,2
5720 INPUT "WERT: "; LINE C$(CC)
5730 LET CC=CC+1: IF CC=41 THEN
LET CC=1
5740 RETURN
5800:
5801:
5810 LET B$="DIODE"
5820 LET HORIZ=5860
5830 LET VERTI=5950
5840 GO TO RICHTUNG
5860 IF XX>232 AND AA<>-1 THEN G
O TO 9990
5870 IF XX<16 AND AA<>1 THEN GO
TO 9990
5875 IF YY<3 OR YY>166 THEN GO T
O 9990
5880 PLOT XX,YY: DRAW AA*16,0
5885 PLOT XX+AA*5,YY-3: DRAW 0,6
5890 DRAW AA*5,-2: PLOT XX+AA*5,
YY-3
5895 DRAW AA*5,2
5900 PLOT XX+AA*6,YY-2: DRAW 0,4
5905 PLOT XX+AA*7,YY-1: DRAW 0,3
5910 PLOT XX+AA*8,YY-1: DRAW 0,2
5920 PLOT XX+AA*11,YY-3: DRAW 0,
6
5922 PRINT #0;AT 1,0;"ZENERDIODE
? --      (JA=5,NEIN=8)     "
5923 IF INKEY$<>"" THEN GO TO 59
23
5924 IF INKEY$="8" THEN GO TO 59
29
5925 IF INKEY$="5" THEN GO TO 59
27
5926 GO TO 5924
5927 PLOT XX+AA*11,YY-3: DRAW -A
A*2,0
5929 LET BB1=3: LET BB2=7: LET B
B3=11: IF AA<>1 THEN LET BB1=-13
: LET BB2=-9: LET BB3=-5
5930 PLOT XX+BB1,YY+5: DRAW 1,0:
 DRAW 1,1: DRAW 0,1: DRAW -1,1:
DRAW -1,0: DRAW 0,-3
5931 LET XXX=XX+BB2: LET YYY=YY+
5: LET M$=STR$ DD: GO SUB 8100+1
0*VAL M$(1)
5932 IF DD>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
5939 GO TO 6050
5950 IF YY>159 AND AA<>-1 THEN G
O TO 9990
5960 IF YY<16 AND AA<>1 THEN GO
TO 9990
5965 IF XX<3 THEN GO TO 9990
5970 PLOT XX,YY: DRAW 0,AA*16
5975 PLOT XX-3,YY+AA*5: DRAW 6,0
5980 DRAW -2,AA*5: PLOT XX-3,YY+
AA*5
5985 DRAW 2,AA*5
5990 PLOT XX-3,YY+AA*11: DRAW 6,
0
6000 PLOT XX-2,YY+AA*6: DRAW 4,0
6005 PLOT XX-1,YY+AA*7: DRAW 3,0
6010 PLOT XX-1,YY+AA*8: DRAW 2,0
6020 PRINT #0;AT 1,0;"ZENERDIODE
? --      (JA=5,NEIN=8)     "
6025 IF INKEY$<>"" THEN GO TO 60
25
6030 IF INKEY$="8" THEN GO TO 60
46
6035 IF INKEY$="5" THEN GO TO 60
45
6040 GO TO 6030
6045 PLOT XX-3,YY+AA*11: DRAW 0,
-AA*2
6046 LET BB1=5: LET BB2=9: LET B
B3=13: IF XX>240 THEN LET BB1=-1
5: LET BB2=-11: LET BB3=-7
6047 PLOT XX+BB1,YY+AA*9: DRAW 1
,0: DRAW 1,1: DRAW 0,1: DRAW -1,
1: DRAW -1,0: DRAW 0,-3
6048 LET XXX=XX+BB2: LET YYY=YY+
AA*9: LET M$=STR$ DD: GO SUB 810
0+10*VAL M$(1)
6049 IF DD>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
6050 INPUT "WERT: "; LINE D$(DD)
6060 LET DD=DD+1: IF DD=41 THEN
LET DD=1
6070 RETURN
6100:
6102:
6110 LET B$="POTENTIOMETER"
6120 LET HORIZ=6160
6130 LET VERTI=6260
6140 GO TO RICHTUNG
6160 IF YY<8 THEN GO TO 9990
6170 IF XX>232 AND AA<>-1 THEN G
O TO 9990
6180 IF XX<16 AND AA<>1 THEN GO
TO 9990
6190 PLOT XX,YY: DRAW AA*3,0
6195 PLOT XX+AA*3,YY+2: DRAW AA*
10,0
6200 PLOT XX,YY: DRAW AA*3,0
6205 PLOT XX+AA*3,YY+2: DRAW AA*
10,0
6210 DRAW 0,-4: DRAW -AA*10,0
6215 DRAW 0,4
6220 PLOT XX+AA*13,YY: DRAW AA*3
,0
6225 PLOT XX+AA*15,YY-8: DRAW -A
A*10,13
6250 DRAW -AA,-1: PLOT XX+AA*5,Y
Y+6
6255 DRAW AA*2,0
6257 GO TO 6295
6263 IF YY>152 AND AA<>-1 THEN G
O TO 9990
6265 IF YY<16 AND AA<>1 THEN GO
TO 9990
6267 IF XX<8 THEN GO TO 9990
6270 PLOT XX,YY: DRAW 0,AA*3
6272 PLOT XX+2,YY+AA*3: DRAW 0,A
A*10
6275 DRAW -4,0: DRAW 0,-AA*10
6277 DRAW 4,0
6280 PLOT XX,YY+AA*13: DRAW 0,AA
*3
6285 PLOT XX-8,YY+AA*15: DRAW 13
,-AA*10
6290 DRAW -1,-AA: PLOT XX+6,YY+A
A*5: DRAW 0,AA*2
6295 INPUT "WERT: "; LINE R$(RR)
6296 LET RR=RR+1: IF RR=41 THEN
LET RR=1
6297 RETURN
6300:
6302:
6305 LET B$="ERDUNG"
6310 LET HORIZ=6335
6315 LET VERTI=6365
6320 GO TO RICHTUNG
6335 IF YY<4 THEN GO TO 9990
6337 IF XX<7 AND AA<>1 THEN GO T
O 9990
6340 PLOT XX,YY: PLOT XX+AA,YY
6342 PLOT XX+AA*2,YY+4: DRAW 0,-
8
6345 PLOT XX+AA*4,YY+3: DRAW 0,-
6
6350 PLOT XX+AA*6,YY+2: DRAW 0,-
4
6355 GO TO 6382
6365 IF YY<7 AND AA<>1 THEN GO T
O 9990
6367 IF XX<4 THEN GO TO 9990
6370 PLOT XX,YY: PLOT XX,YY+AA
6372 PLOT XX+4,YY+AA*2: DRAW -8,
0
6375 PLOT XX+3,YY+AA*4: DRAW -6,
0
```

Basic-Listing »Platinen-CAD« (Fortsetzung)

```
6380 PLOT XX+2,YY+AA*6: DRAW -4,
0
6382 LET E$(EE)=B$
6385 LET EE=EE+1: IF EE=41 THEN
LET EE=1
6387 RETURN
6390:
6392:
6395 LET B$="UEBERTRAGER"
6400 LET HORIZ=6415
6405 LET VERTI=6435
6410 GO TO RICHTUNG
6415 IF XX>232 AND AA<>-1 THEN G
O TO 9990
6417 IF XX<16 THEN IF AA<>1 THEN
 GO TO 9990
6418 IF YY>152 THEN GO TO 9990
6420 PLOT XX,YY: PLOT XX+AA,YY:
DRAW AA*5,0: DRAW 0,16: DRAW -AA
,0
6422 DRAW 0,-16: DRAW -AA,0: DRA
W 0,16: DRAW -AA*4,0
6425 PLOT XX+AA*15,YY: DRAW -AA*
5,0: DRAW 0,16: DRAW AA,0
6427 DRAW 0,-16: DRAW AA,0: DRAW
 0,16: DRAW AA*4,0
6430 GO TO 6450
6436 IF YY>152 THEN IF AA<>-1 TH
EN GO TO 9990
6437 IF YY<16 THEN IF AA<>1 THEN
 GO TO 9990
6438 IF XX>232 THEN GO TO 9990
6440 PLOT XX,YY: PLOT XX,YY+AA:
DRAW 0,AA*5: DRAW 16,0: DRAW 0,-
AA
6442 DRAW -16,0: DRAW 0,-AA: DRA
W 16,0: DRAW 0,-AA*4
6445 PLOT XX,YY+AA*16: DRAW 0,-A
A*5: DRAW 16,0: DRAW 0,AA
6447 DRAW -16,0: DRAW 0,AA: DRAW
 16,0: DRAW 0,AA*4
6450 LET E$(EE)=B$
6455 LET EE=EE+1: IF EE=41 THEN
LET EE=1
6457 RETURN
6460:
6462:
6465 LET B$="TRANSISTOR"
6480 LET HORIZ=6500
6485 LET VERTI=6560
6490 GO TO RICHTUNG
6500 IF XX<16 AND AA<>1 THEN GO
TO 9990
6505 IF XX>232 AND AA<>-1 THEN G
O TO 9990
6507 IF YY>160 OR YY<8 THEN GO T
O 9990
6510 CIRCLE XX+7*AA,YY,7
6515 PLOT XX,YY: DRAW AA*6,0
6520 PLOT XX+7*AA,YY+4: DRAW 0,-
8
6525 PLOT XX+8*AA,YY+1: DRAW AA,
0: DRAW AA*6,6
6530 PLOT XX+8*AA,YY-1: DRAW AA*
7,-7
6535 PRINT #0;AT 1,0;"NPN ODER P
NP-TRAN.?(NPN=5,PNP=8)"
6540 IF INKEY$<>"" THEN GO TO 65
40
6545 IF INKEY$="5" THEN PLOT XX+
AA*8,YY-4: DRAW AA*3,0: DRAW 0,3
: PLOT XX+AA*9,YY-3: DRAW AA,AA:
 GO TO 6552
6550 IF INKEY$="8" THEN PLOT XX+
AA*9,YY-2: DRAW 0,-3: DRAW AA*3,
3: DRAW -AA*3,0: GO TO 6552
6551 GO TO 6545
6552 LET BB1=3: LET BB2=6: LET B
B3=10: IF AA<>1 THEN LET BB1=-11
: LET BB2=-8: LET BB3=-4
6553 PLOT XX+BB1,YY+9: DRAW 0,3:
 DRAW -1,0: DRAW 2,0
6554 LET XXX=XX+BB2: LET YYY=YY+
9: LET M$=STR$ TT: GO SUB 8100+1
0*VAL M$(1)
6555 IF TT>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
6559 GO TO 6625
6565 IF YY>152 AND AA<>-1 THEN G
O TO 9990
6570 IF YY<16 AND AA<>1 THEN GO
TO 9990
6572 IF XX<8 OR XX>240 THEN GO T
O 9990
6575 CIRCLE XX,YY+7*AA,7
6580 PLOT XX,YY: DRAW 0,AA*6
6585 PLOT XX+4,YY+7*AA: DRAW -8,
0
6590 PLOT XX+1,YY+8*AA: DRAW 0,A
A: DRAW 6,AA*6
6595 PLOT XX-1,YY+8*AA: DRAW -7,
AA*7
6600 PRINT #0;AT 1,0;"NPN ODER P
NP-TRAN.?(NPN=5,PNP=8)"
6605 IF INKEY$<>"" THEN GO TO 66
05
6610 IF INKEY$="5" THEN PLOT XX-
4,YY+AA*8: DRAW 0,AA*3: DRAW 3,0
: PLOT XX-3,YY+AA*9: DRAW AA,AA:
 GO TO 6617
6615 IF INKEY$="8" THEN PLOT XX-
2,YY+AA*9: DRAW -3,0: DRAW 3,AA*
3: DRAW 0,-AA*3: GO TO 6617
6616 GO TO 6610
6617 LET BB1=10: LET BB2=13: LET
 BB3=17: IF XX>236 THEN LET BB1=
-18: LET BB2=-15: LET BB3=-11
6618 PLOT XX+BB1,YY+AA*5: DRAW 0
,3: DRAW -1,0: DRAW 2,0
6619 LET XXX=XX+BB2: LET YYY=YY+
AA*5: LET M$=STR$ TT: GO SUB 810
0+10*VAL M$(1)
6620 IF TT>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
6625 INPUT "WERT(TYP): "; LINE T
$(TT)
6630 LET TT=TT+1: IF TT=41 THEN
LET TT=1
6635 RETURN
6650:
6652:
6655 PRINT AT 21-INT (YY/8),INT
(XX/8);" "
6660 RETURN
6670:
6672:
6675 LET B$="SPULE"
6680 LET HORIZ=6700
6685 LET VERTI=6745
6690 GO TO RICHTUNG
6700 IF XX>232 AND AA<>-1 THEN G
O TO 9990
6705 IF XX<16 AND AA<>1 THEN GO
TO 9990
6710 PLOT XX,YY: DRAW AA*2,0
6715 DRAW AA*2,2: DRAW AA*2,-2
6720 DRAW AA*2,2: DRAW AA*2,-2
6725 DRAW AA*2,2: DRAW AA*2,-2
6730 DRAW AA*2,0
6731 LET BB1=5: LET BB2=7: LET B
B3=11: IF AA<>1 THEN LET BB1=-11
: LET BB2=-9: LET BB3=-5
6732 PLOT XX+BB1,YY+4: DRAW -2,0
: DRAW 0,3
6733 LET XXX=XX+BB2: LET YYY=YY+
4-(YY=168): LET M$=STR$ LL: GO S
UB 8100+10*VAL M$(1)
6734 IF LL>9 THEN LET XXX=XX+BB3
: GO SUB 8100+10*VAL M$(2)
6739 GO TO 6780
6745 IF YY>152 AND AA<>-1 THEN G
O TO 9990
6750 IF YY<16 AND AA<>1 THEN GO
TO 9990
6755 PLOT XX,YY: DRAW 0,AA*2
6760 DRAW 2,AA*2: DRAW -2,AA*2
6765 DRAW 2,AA*2: DRAW -2,AA*2
6770 DRAW 2,AA*2: DRAW -2,AA*2
6775 DRAW 0,AA*2
6776 LET BB1=7: LET BB2=9: LET B
B3=13: IF XX>240 THEN LET BB1=-1
0: LET BB2=-8: LET BB3=-4
6777 PLOT XX+BB1,YY+AA*9: DRAW -
2,0: DRAW 0,3
6778 LET XXX=XX+BB2: LET YYY=YY+
AA*9: LET M$=STR$ LL: GO SUB 810
0+10*VAL M$(1)
6779 IF LL>9 THEN LET XXX=XX+BB3
```

```
  GO SUB 8100+10*VAL M$(2)
6780 INPUT "WERT: "; LINE L$(LL)
6785 LET LL=LL+1: IF LL=41 THEN
LET LL=1
6790 RETURN
6800:
6802:
6810 LET B$="BATTERIE"
6820 LET HORIZ=6855
6830 LET VERTI=6920
6840 GO TO RICHTUNG
6855 IF XX<16 AND AA<>1 THEN GO
TO 9990
6860 IF XX>232 AND AA<>-1 THEN G
O TO 9990
6863 IF YY<5 THEN GO TO 9990
6865 PLOT XX,YY: DRAW AA*5,0
6870 PLOT XX+AA*6,YY+3: DRAW 0,-
6
6875 DRAW AA,0: DRAW 0,6
6880 PLOT XX+AA*9,YY+5: DRAW 0,-
10
6885 PLOT XX+AA*10,YY: DRAW AA*6
,0
6890 PLOT XX+AA*2,YY+3: DRAW AA*
2,0
6895 PLOT XX+AA*11,YY+3: DRAW AA
*2,0
6900 PLOT XX+AA*12,YY+4: DRAW 0,
-2
6910 GO TO 6975
6925 IF YY>152 AND AA<>-1 THEN G
O TO 9990
6930 IF YY<16 AND AA<>1 THEN GO
TO 9990
6933 IF XX<5 THEN GO TO 9990
6935 PLOT XX,YY: DRAW 0,AA*5
6940 PLOT XX+3,YY+AA*6: DRAW -6,
0
6945 DRAW 0,AA: DRAW 6,0
6950 PLOT XX+5,YY+AA*9: DRAW -10
,0
6955 PLOT XX,YY+AA*10: DRAW 0,AA
*6
6960 PLOT XX+3,YY+AA*2: DRAW 0,A
A*2
6965 PLOT XX+3,YY+AA*11: DRAW 0,
AA*2
6970 PLOT XX+4,YY+AA*12: DRAW -2
,0
6975 LET E$(EE)=B$
6980 LET EE=EE+1: IF EE=41 THEN
LET EE=1
6985 RETURN
7000:
7002:
7010 LET B$="GLUEHLAMPE"
7020 LET HORIZ=7060
7030 LET VERTI=7110
7040 GO TO RICHTUNG
7060 IF XX>232 AND AA<>-1 THEN G
O TO 9990
7070 IF XX<16 AND AA<>1 THEN GO
TO 9990
7073 IF YY<4 THEN GO TO 9990
7075 PLOT XX,YY: DRAW AA*4,0
7080 PLOT XX+AA*10,YY-2: DRAW -A
A*4,4
7085 PLOT XX+AA*6,YY-2: DRAW AA*
4,4
7090 PLOT XX+AA*12,YY: DRAW AA*4
,0
7095 CIRCLE XX+AA*8,YY,4
7100 GO TO 7160
7120 IF YY>152 AND AA<>-1 THEN G
O TO 9990
7130 IF YY<16 AND AA<>1 THEN GO
TO 9990
7133 IF XX<3 THEN GO TO 9990
7135 PLOT XX,YY: DRAW 0,AA*4
7140 PLOT XX-2,YY+AA*10: DRAW 4,
-AA*4
7145 PLOT XX-2,YY+AA*6: DRAW 4,A
A*4
7150 PLOT XX,YY+AA*12: DRAW 0,AA
*4
7155 CIRCLE XX,YY+AA*8,4
7160 LET E$(EE)=B$
7165 LET EE=EE+1: IF EE=41 THEN
LET EE=1
7170 RETURN
7200:
7201:
7205 IF YY<1 OR XX<1 THEN GO TO
9990
7210 CIRCLE XX,YY,1.4
7220 RETURN
7250:
7251:
7255 LET B$="LAUTSPRECHER"
7260 LET HORIZ=7280
7265 LET VERTI=7340
7270 GO TO RICHTUNG
7285 IF XX<8 AND AA<>1 THEN GO T
O 9990
7290 IF XX<2 THEN GO TO 9990
7295 IF YY>160 THEN GO TO 9990
7300 PLOT XX-AA*2,YY+2: DRAW 0,3
7305 DRAW AA*5,0: DRAW 0,-3
7310 DRAW AA*2,-2: DRAW 0,7
7315 DRAW -AA*2,-2: DRAW 0,-3
7320 DRAW -AA*5,0: PLOT XX,YY
7325 DRAW 0,2: PLOT XX,YY+5
7330 DRAW 0,2
7335 GO TO 7390
7345 IF YY<8 AND AA<>1 THEN GO T
O 9990
7350 IF YY<2 THEN GO TO 9990
7352 IF XX>240 THEN GO TO 9990
7355 PLOT XX+2,YY-AA*2: DRAW 3,0
7360 DRAW 0,5*AA: DRAW -3,0
7365 DRAW -2,AA*2: DRAW 7,0
7370 DRAW -2,-AA*2: DRAW -3,0
7375 DRAW 0,-AA*5: PLOT XX,YY
7380 DRAW 2,0: PLOT XX+5,YY
7385 DRAW 2,0
7390 LET E$(EE)=B$
7395 LET EE=EE+1: IF EE=41 THEN
LET EE=1
7400 RETURN
7410:
7412:
7415 LET B$="MIKROFON"
7420 LET HORIZ=7435
7425 LET VERTI=7480
7430 GO TO RICHTUNG
7440 IF XX<16 AND AA<>1 THEN GO
TO 9990
7445 IF XX>240 AND AA<>-1 THEN G
O TO 9990
7447 IF YY>160 OR YY<1 THEN GO T
O 9990
7450 CIRCLE XX+AA*11,YY+4,4: PLO
T XX+AA*15,YY-1
7455 DRAW 0,10: PLOT XX,YY+1
7460 DRAW 0,1: DRAW AA*7,0
7465 PLOT XX,YY+7: DRAW 0,-2
7470 DRAW AA*7,0
7475 RETURN
7485 IF YY<16 AND AA<>1 THEN GO
TO 9990
7486 IF YY>160 AND AA<>-1 THEN G
O TO 9990
7490 IF XX<1 OR XX>240 THEN GO T
O 9990
7500 CIRCLE XX+3,YY+AA*11,4: PLO
T XX-1,YY+AA*15
7505 DRAW 9,0: PLOT XX+1,YY
7510 DRAW 1,0: DRAW 0,AA*7
7515 PLOT XX+7,YY: DRAW -2,0
7520 DRAW 0,AA*7
7525 LET E$(EE)=B$
7530 LET EE=EE+1: IF EE=41 THEN
LET EE=1
7535 RETURN
7600:
7602:
7670 REM BILD LOESCH.
7675 PRINT #0;AT 0,0;" WOLLEN SI
E WIRKLICH DAS GANZE           BI
LD LOESCHEN ??      "
7680 IF INKEY$<>"" THEN GO TO 76
80
7682 IF INKEY$="0" THEN GO SUB 9
990: GO TO 9890
7685 IF INKEY$="J" OR INKEY$="j"
 THEN CLS : GO TO 9890
7690 IF INKEY$="N" OR INKEY$="n"
```

Basic-Listing »Platinen-CAD« (Fortsetzung)

```
 THEN GO SUB 9990: GO TO 9890
7695 GO TO 7682
7696:
7697:
7700 REM LISTE LOESCH.
7710 PRINT #0;AT 0,0;"WELCHE LIS
TE MOECHTEN SIE VOLL- STAENDIG L
OESCHEN? (R,T,C,D,L,E)"
7715 IF INKEY$<>"" THEN GO TO 77
15
7720 LET Z$=INKEY$
7722 IF Z$="" THEN GO TO 7720
7725 IF LEN Z$<>1 THEN GO TO 772
0
7730 IF Z$="R" THEN LET RR=1: DI
M R$(41,8): GO TO 7765
7735 IF Z$="T" THEN LET TT=1: DI
M T$(41,8): GO TO 7765
7740 IF Z$="D" THEN LET DD=1: DI
M D$(41,8): GO TO 7765
7745 IF Z$="C" THEN LET CC=1: DI
M C$(41,8): GO TO 7765
7750 IF Z$="L" THEN LET LL=1: DI
M L$(41,8): GO TO 7765
7755 IF Z$="E" THEN LET EE=1: DI
M E$(41,12): GO TO 7765
7757 IF Z$="0" THEN PLOT OVER 1,
XX,YY: GO TO 9890
7760 GO TO 7710
7765 PRINT #0;AT 0,0;Z$;"  WIRD
GELOESCHT-TASTE DRUECKEN";AT 1,0
;Y$
7767 PAUSE 0
7770 PLOT OVER 1,XX,YY: GO TO 98
90
7771:
7772:
7780 REM EINZELTEIL LOESCH.
7781 INPUT "NUMMER DES BAUTEILS
? ";NU: IF NU<1 OR NU>40 THEN GO
 TO 7781
7785 PRINT #0;AT 0,0;"   BAUTEIL
 ?(R,T,D,C,L ODER E)   "
7790 IF INKEY$<>"" THEN GO TO 77
90
7795 LET Z$=INKEY$
7800 IF LEN Z$<>1 THEN GO TO 779
5
7802 IF Z$="0" THEN GO TO 7770
7805 IF Z$="R" THEN LET R$(NU)="
": GO TO 7840
7810 IF Z$="T" THEN LET T$(NU)="
": GO TO 7840
7815 IF Z$="D" THEN LET D$(NU)="
": GO TO 7840
7820 IF Z$="C" THEN LET C$(NU)="
": GO TO 7840
7825 IF Z$="L" THEN LET L$(NU)="
": GO TO 7840
7830 IF Z$="E" THEN LET E$(NU)="
": GO TO 7840
7835 GO TO 7785
7840 PRINT #0;AT 0,0;Z$;NU;" IST
 GELOESCHT-TASTE DRUECKEN   ";AT
1,0;Y$
7845 PAUSE 0: PLOT OVER 1,XX,YY:
 GO TO 9890
7846:
7847:
7850 REM LISTEN SAVEN
7855 PRINT #0;AT 0,0;"WOLLEN SIE
 DIE BAUTEILE LADEN    ODER SAVEN
?    (LADEN=5,SAVEN=8)   "
7858 IF INKEY$<>"" THEN GO TO 78
58
7859 IF INKEY$="0" THEN GO SUB 9
990: GO TO 9890
7860 IF INKEY$="5" THEN GO TO 82
00
7862 IF INKEY$="8" THEN GO TO 78
64
7863 GO TO 7859
7864 PRINT #0;AT 0,0;"SO OFT STA
RT TAPE THEN... KOMMT TIPPEN SIE
 ENTER !     <ENTER>   ": PAUSE 0
7865 PRINT #0;AT 0,0;"DIE AUFNAH
ME BEGINNT NUN !         "
7870 LET R$(41)=STR$ RR: SAVE "W
IDERSTAND" DATA R$(): LET T$(41)
=STR$ TT: SAVE "TRANSISTOR" DATA
 T$(): LET C$(41)=STR$ CC: SAVE
"KONDENSATO" DATA C$()
7875 LET D$(41)=STR$ DD: SAVE "D
IODE" DATA D$(): LET L$(41)=STR$
 LL: SAVE "SPULE" DATA L$(): LET
 E$(41)=STR$ EE: SAVE "DIVERSE"
DATA E$()
7880 PRINT #0;AT 0,0;"SPULEN SIE
 ZURUECK UND SCHALTEN SIE AUF WI
EDERGABE !     <ENTER>   ": PAUSE 0
7881 PRINT #0;AT 0,0;"TRITT Tape
 loading Error AUF,SO TIPPEN SIE
 GOTO 4000 ! <ENTER>   ": PAUSE 0
7883 PRINT #0;AT 0,0;"DARAUFHIN
WIEDERHOLEN SIE DIESE ROUTINE NO
CHEINMAL !     <ENTER>   ": PAUSE 0
7885 GO SUB 9890
7890 PRINT AT 0,0: VERIFY "" DAT
A R$(): PRINT AT 0,0: VERIFY ""
DATA T$(): PRINT AT 0,0: VERIFY
"" DATA C$()
7895 PRINT AT 0,0: VERIFY "" DAT
A D$(): PRINT AT 0,0: VERIFY ""
DATA L$(): PRINT AT 0,0: VERIFY
"" DATA E$()
7896 PRINT #0;AT 0,0;"AUFNAHME I
ST KORREKT VERLAUFEN. BITTE DREU
CKEN SIE ENTER.      "
7897 PAUSE 0: GO TO 9890
7898:
7899:
7900 REM HARDCOPY
7915 PRINT #0;AT 0,0;"  WOLLEN S
IE EINE HARDCOPY DES     BILDES
?    (JA=8,NEIN=5)   "
7920 IF INKEY$<>"" THEN GO TO 79
20
7925 IF INKEY$="8" THEN GO SUB 9
990: GO TO 9890
7930 IF INKEY$="5" THEN GO TO 79
40
7935 GO TO 7925
7940 COPY : POKE 23300,192: POKE
 23301,23
7945 GO SUB 9990: GO TO 9890
7946:
7947:
7950 REM LISTENAUSDR.
7955 PRINT #0;AT 0,0;"WOLLEN SIE
 DIE BAUTEILLISTE AUS-   DRUCKEN
 ? --   (JA=5,NEIN=8)   "
7960 IF INKEY$<>"" THEN GO TO 79
60
7965 IF INKEY$="8" THEN GO SUB 9
990: GO TO 9890
7970 IF INKEY$="5" THEN GO TO 79
80
7975 GO TO 7965
7980 GO SUB 9890: LPRINT : LPRIN
T : LPRINT
7981 LPRINT "WIDERSTAENDE";TAB 1
6;"KONDENSATOREN": LPRINT
7982 LET N=1: LET X=1
7983 LET W$="": LET V$=""
7984 IF R$(N)<>Y$( TO 10) AND N<
RR THEN LET W$="R"+STR$ N+"  "+R
$(N): GO TO 7986
7985 IF N<RR THEN GO SUB 7993: G
O TO 7984
7986 IF C$(X)<>Y$( TO 10) AND X<
CC THEN LET V$="C"+STR$ X+"  "+C
$(X): GO TO 7988
7987 IF X<CC THEN GO SUB 7994: G
O TO 7986
7988 LPRINT W$;TAB 16;V$
7989 IF N<RR THEN GO SUB 7993
7990 IF X<CC THEN GO SUB 7994
7991 IF N=RR AND X=CC THEN GO TO
 8000
7992 GO TO 7983
7993 LET N=N+1: RETURN
7994 LET X=X+1: RETURN
8000 LPRINT : LPRINT : LPRINT
8005 LPRINT "DIODEN";TAB 16;"TRA
NSISTOREN": LPRINT
8006 LET N=1: LET X=1
8007 LET W$="": LET V$=""
8008 IF D$(N)<>Y$( TO 10) AND N<
DD THEN LET W$="D"+STR$ N+"  "+D
```

```
$(N): GO TO 8010
8009 IF N<DD THEN GO SUB 7993: G
O TO 8008
8010 IF T$(X)<>Y$( TO 10) AND X<
TT THEN LET V$="T"+STR$ X+"   "+T
$(X): GO TO 8012
8011 IF X<TT THEN GO SUB 7994: G
O TO 8010
8012 LPRINT W$;TAB 16;V$
8013 IF N<DD THEN GO SUB 7993
8014 IF X<TT THEN GO SUB 7994
8015 IF N=DD AND X=TT THEN GO TO
 8020
8017 GO TO 8007
8020 LPRINT : LPRINT : LPRINT
8025 LPRINT "SPULEN";TAB 16;"DIV
ERSE": LPRINT
8026 LET N=1: LET X=1
8027 LET W$="": LET V$=""
8028 IF L$(N)<>Y$( TO 10) AND N<
LL THEN LET W$="L"+STR$ N+"   "+L
$(N): GO TO 8030
8029 IF N<LL THEN GO SUB 7993: G
O TO 8028
8030 IF E$(X)<>Y$( TO 10) AND N<
EE THEN LET V$=E$(X): GO TO 8035
8031 IF X<EE THEN GO SUB 7994: G
O TO 8030
8035 LPRINT W$;TAB 16;V$
8036 IF N<LL THEN GO SUB 7993
8037 IF X<EE THEN GO SUB 7994
8038 IF N=LL AND X=EE THEN GO TO
 8040
8039 GO TO 8027
8040 POKE 23300,192: POKE 23301,
23
8045 GO SUB 9990: GO TO 9890
8046:
8047:
8050 REM SCREEN SAVE
8055 PRINT #0;AT 0,0;"WOLLEN SIE
 DEN BILDSCHIRM SAVEN ODER LADEN
 ? (SAVEN=5,LADEN=8)        "
8060 IF INKEY$<>"" THEN GO TO 80
60
8065 IF INKEY$="0" THEN GO SUB 9
990: GO TO 9890
8066 IF INKEY$="5" THEN GO TO 80
70
8067 IF INKEY$="8" THEN GO TO 80
86
8068 GO TO 8065
8070 PRINT #0;AT 0,0;"BEREITEN S
IE DIE AUFNAHME VOR !
   <ENTER>              ": PAUSE 0
8075 GO SUB 9890: PRINT #0;AT 0,
0
8080 GO SUB 9990: SAVE "BILD"COD
E 16384,6080
8085 GO SUB 9990: GO TO 9890
8086 REM LADEN
8087 PRINT #0;AT 0,0;"SCHALTEN S
IE AUF WIEDERGABE !
                          "
8090 LOAD ""CODE 16384: GO SUB 9
990
8091 GO TO 8085
8092:
8093:
8094:
8100 REM ZAHLEN
8102 PLOT XXX+1,YYY: DRAW 1,1: D
RAW 0,2: DRAW -1,1: DRAW -1,-1:
DRAW 0,-2
8105 RETURN
8112 PLOT XXX+2,YYY: DRAW 0,4: D
RAW -2,-2
8115 RETURN
8122 PLOT XXX+2,YYY: DRAW -2,0:
DRAW 0,1: DRAW 2,2: DRAW 0,1: DR
AW -2,0
8125 RETURN
8132 PLOT XXX,YYY: DRAW 2,0: DRA
W 0,1: DRAW -1,1: DRAW 1,1: DRAW
 0,1: DRAW -2,0
8135 RETURN
8142 PLOT XXX+2,YYY: DRAW 0,4: D
RAW -2,-2: DRAW 0,-1: DRAW 2,0
8145 RETURN
8152 PLOT XXX,YYY: DRAW 2,0: DRA
W 0,2: DRAW -1,0: DRAW -1,1: DRA
W 0,1: DRAW 2,0
8155 RETURN
8162 PLOT XXX+2,YYY+4: DRAW -1,0
: DRAW -1,-1: DRAW 0,-3: DRAW 2,
0: DRAW 0,2: DRAW -2,0
8165 RETURN
8172 PLOT XXX,YYY: DRAW 0,1: DRA
W 2,2: DRAW 0,1: DRAW -2,0
8175 RETURN
8182 PLOT XXX,YYY: DRAW 2,0: DRA
W 0,1: DRAW -2,2: DRAW 0,1: DRAW
 2,0: DRAW 0,-1: DRAW -2,-2: DRA
W 0,-1
8185 RETURN
8192 PLOT XXX,YYY: DRAW 2,0: DRA
W 0,4: DRAW -2,0: [illegible]
W 2,0
8195 RETURN
8196:
8197:
8200 REM LISTEN LADEN
8205 PRINT #0;AT 0,0;"LEGEN SIE
DIE CASSETTE EIN UND  SCHALTEN S
IE AUF WIEDERGABE !     "
8210 LOAD "" DATA R$(): LOAD ""
DATA C$(): LOAD "" DATA D$(): LO
AD "" DATA T$(): LOAD "" DATA L$
(): LOAD "" DATA E$()
8215 GO SUB 9990: GO TO 9890
8220:
8230:
8300 REM M.-CODE
8310 DATA 42,0,91,237,91,2,91,23
7,75,4,91,120,177,200,167,237,82
,200,25,56,3,237,176,201,235,9,2
35,9,43,27,237,184,201
8320 FOR N=65503 TO 65535: READ
MAS: POKE N,MAS: NEXT N
8330 RETURN
8350 REM UM
8356 POKE 23296,0: POKE 23297,64
: POKE 23298,93: POKE 23299,208:
 RANDOMIZE USR 65500: POKE 23296
,30: POKE 23297,232: POKE 23298,
0: POKE 23299,64: RANDOMIZE USR
65500
8360 RETURN
8366 POKE 23296,0: POKE 23297,64
: POKE 23298,30: POKE 23299,232:
 RANDOMIZE USR 65500: POKE 23296
,93: POKE 23297,208: POKE 23298,
0: POKE 23299,64: RANDOMIZE USR
65500
8370 RETURN
8405 IF EI=0 THEN GO SUB 9990: G
O SUB 8355: LET EI=1: GO TO 8415
8410 IF EI=1 THEN GO SUB 9990: G
O SUB 8365: LET EI=0
8415 RETURN
8500:
8502:
9800 REM RICHTUNGSABFRAGE
9810 PRINT #0;AT 1,0;" RICHTUNG
MIT CURSOR WAEHLEN !  "
9820 PRINT #0;AT 0,(31-LEN B$)/2
; FLASH 1;B$
9830 LET Z$=INKEY$
9835 IF INKEY$="0" THEN GO SUB 9
990: GO TO 9890
9840 IF Z$="5" THEN LET AA=-1: G
O SUB HORIZ: GO TO 9890
9850 IF Z$="8" THEN LET AA=1: GO
 SUB HORIZ: GO TO 9890
9860 IF Z$="6" THEN LET AA=-1: G
O SUB VERTI: GO TO 9890
9870 IF Z$="7" THEN LET AA=1: GO
 SUB VERTI: GO TO 9890
9880 GO TO 9830
9890 PRINT #0;AT 0,0;Y$
9900 PRINT #0;AT 1,0;Y$
9910 RETURN
9911:
9912:
9990 REM PLOT-KORREKT.
9991 PLOT OVER 1,XX,YY
9992 RETURN
9999 SAVE "PLATI.-CAD" LINE 1000
```

Basic-Listing »Platinen-CAD« (Schluß)

Internas aus dem Spectrum

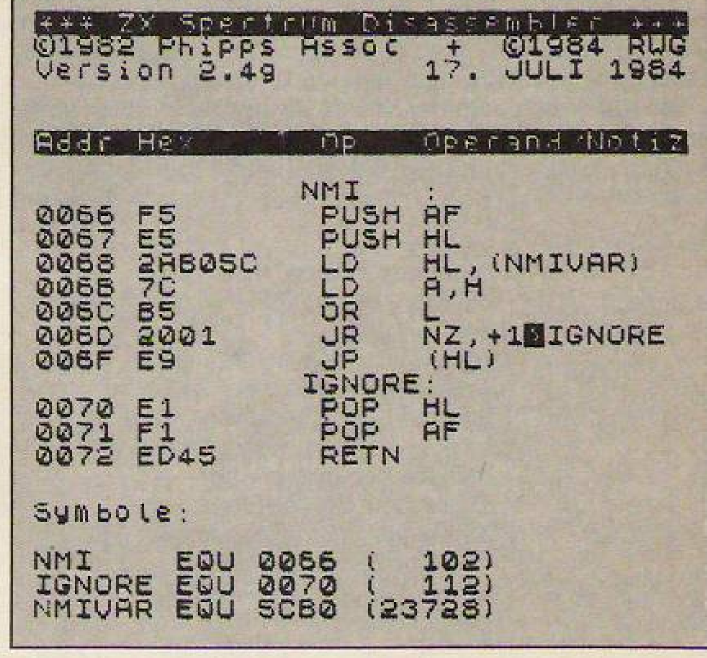
Zwei wertvolle Tips, die den nicht markierbaren Interrupt und den freien Speicherplatz betreffen.

In Happy-Computer, Ausgabe 11/1984, wurde gezeigt, wie man den maskierbaren Interrupt, also eine abschaltbare Unterbrechung, benutzt. Aber auch der nicht maskierbare Interrupt (NMI) kann genutzt werden. Der entsprechende Anschluß des Z80A-Mikroprozessors, ist zwar lediglich an den Userport geführt, aber das reicht aus. Wenn man »RANDOMIZE USR 102« (102 dez. = 66 hex.) eintippt, gibt es einen Systemkaltstart. Ist das nur eine Alternative zu »RANDOMIZE USR 0« dem »offiziellen« Systemkaltstart? Jetzt tippen wir »POKE 23728,1« und dann »RANDOMIZE USR 102«. Nun kommt die Meldung »o.k.« Schauen wir uns das Listing an. Es zeigt die Routine an der Adresse 66H. Wir sehen, daß der Wert von NMIVAR geprüft wird. Ist er Null gibt es einen Kaltstart, ist er < > Null, wird zurückgesprungen und mit dem normalen Programm weitergemacht. (Stünde da der Sprungbefehl »JR Z,+1«, dann wäre es fantastisch. Man könnte durch den nichtmaskierbaren Interrupt an eine beliebige Stelle springen. »Wann immer sich der Spectrum aufhängt, könnte man sich retten. Wer also sein ROM in ein EPROM umschießt: diese Stelle bitte ändern.

Wo die Systemvariable NMIVAR ist? Im Handbuch auf Seite 176 steht für Adresse 23728/9: nicht benutzt. Bitte durchstreichen und NMIVAR hinschreiben. Wenn wir jetzt am Userport die Anschlüße OV und NMI (siehe Happy-Computer, Sinclair Sonderheft, Seite 27) mit einem Schalter verbinden, haben wir einen Reset-Schalter, den wir softwaremäßig (»POKE 23728,1« oder »POKE 23728,0«) beeinflussen können. Beim Interface 1 gibt es keine NMI-Routine. Tritt ein NMI auf, während das neue ROM eingeschaltet ist, wird er ignoriert.

Beim Disassemblieren des Spectrum-ROMs fällt noch etwas auf: Obwohl das Spectrum-Basic den Befehl FREE nicht kennt, gibt es die entsprechende Maschinen-Code-Routine. So benutzt man sie: »PRINT 65536-USR 7962«. Diese Methode ist besser als alle anderen. Sie liefert immer den kleinsten Wert. Der Wert ist: SP-(STKEND+80) (siehe Speicherkarte, Spectrumhandbuch, Seite 165). Diese FREE-Routine benutzt den gleichen Test, den der Spectrum macht, um festzustellen, ob noch Speicherplatz vorhanden ist, wenn er welchen benötigt. Die »80«, das sind 80 Byte »eiserne Reserve«, die der Spectrum beim Testen auf freien Speicherplatz immer zurückhält. Noch eine Bemerkung für Profis: Die Variable SP ist der Stackpointer, ein internes Z80A-Register, das von Basic aus sonst nicht zugänglich ist. Das Ergebnis des Aufrufs von USR 7962 liefert den freien Platz als negative Zahl in Zweier-Komplement-Darstellung. »65536-USR 7962« formt diese »negative« Zahl in eine positive um. Falls die Routine den Wert —1 liefern würde, gibt es die Meldung »OUT OF MEMORY«.

(R. W. Gerling)



```
+++ ZX Spectrum Disassembler +++
©1982 Phipps Assoc  +  ©1984 RWG
Version 2.49          17. JULI 1984

Addr Hex          Op   Operand Notiz

                 NMI   :
0066 F5           PUSH AF
0067 E5           PUSH HL
0068 2AB05C       LD   HL,(NMIVAR)
006B 7C           LD   A,H
006C B5           OR   L
006D 2001         JR   NZ,+1 IGNORE
006F E9           JP   (HL)
                 IGNORE:
0070 E1           POP  HL
0071 F1           POP  AF
0072 ED45         RETN

Symbole:

NMI     EQU 0066 (  102)
IGNORE  EQU 0070 (  112)
NMIVAR  EQU 5CB0 (23728)
```

Dieses Listing zeigt die NMI-Maschinencode-Routine zur Abarbeitung des nichtmaskierbaren Interrupts.

Border-Effekte

Mit Hilfe des Programms »Border-Effekt« kann man Border-Effekte auswählen, auflisten und somit in eigenen Programmen verwenden.

Das Programm arbeitet mit dem OUT-Befehl, mit dem man unzählige Border-Effekte entwickeln kann. Dieses menügesteuerte Programm bietet also nur eine Auswahl der Effekte. Die einzelnen Effekte sind im Menü angegeben und eine Beschreibung ist im Programm integriert, so daß hier nur noch die wichtigsten Punkte aufgeführt werden:

Beim Listen eines Border-Effekts muß nach Erscheinen von SCROLL? »n« und dann GO TO 9050 eingegeben werden. Um Border-Effekte mit Ton zu erhalten, muß bei der Eingabe der PAPER-Farbe der jeweilige Wert um 16 erhöht werden.

Die aufgelisteten Border-Effekte können durch einen angeschlossenen Drucker ausgegeben werden. Mit GOTO 9900 kann das Programm »Border-Effekte« auf Band gesichert werden.

(B. Baran)

Variablenliste

s	Dauer des Effekts
a,b,c,d, 1.,2.,3.,4.	Ink-Farbe des Streifens 0-7, keine Farbe: —1
e	Paper-Farbe 0-7, 16-32 mit Ton
x	Schleifenvariable für Border-Effekt
l	Zeile, ab der gelöscht wird; bei Listen
r	Zeile, zu der zurückgesprungen wird
p	Zeile, ab der gelistet wird
u	zu welchem Border-Effekt es geht
p	Zählvariable beim Eingeben der 4 Inkf.
i	Ink-Farbe beim Eingeben
g	zusätzliche Schleifenvariable
a$	Tastatureingabe

Aufschlüsselung der Zeilennummern

30 — 55	:Erklärung ja oder nein?
58 — 95	:Erklärung
100 — 180	:Menü
210 — 224	:Dauer des Effekts wird eingegeben
225 — 260	:Ink-Farben werden eingegeben
270 — 290	:Paper-Farbe wird eingegeben
310 — 340	:Ink-Farbe wird eingegeben
1000 — 1560	:Border-Effekte siehe Listing
9000 — 9050	:Border-Effekt wird gelistet
9100 — 9150	:Ausgabe auf Drucker
9900 — 9930	:SAVEn für Border-Effekte

```
  10 REM  © Bernhard Baran
            Postfach 35
            6710 Frankenthal
            Tel.:06233/24243
  20 BORDER 5: PAPER 5: INK 0: C
LS
  25 PRINT FLASH 1;"          BOR
DER-EFFEKTE          "
  30 PRINT AT 11,0;"Erklaerung n
oetig             (j oder n ei
ngeben)?"
  40 IF INKEY$="j" THEN GO TO 60
  50 IF INKEY$="n" THEN GO TO 10
0
  55 GO TO 40
  58 REM █ Erklaerung █
  60 CLS : PRINT FLASH 1;"
    Erklaerung              "
  70 PRINT AT 2,0;"Border-Effekt
e ist ein Hilfs-  programm, bei
 dem man sich einengewuenschten
Border-Effekt an-  sehen und dan
n auch listen kann.Somit kann ma
n diesen Border-   Effekt leicht
 in eigene         Programme ein
setzen."
  75 PRINT "Beim Listen muss man
 nach dem    Erscheinen von SCROL
L?, n         eingeben, um dann GO
 TO 9050 zu entern."
  80 PRINT "Nachdem man einen Me
nuepunkt      ausgewaehlt hat, mus
s man ver-   schiedene Werte eing
eben:"
  90 PRINT "Bei den Inkfarben ge
lten die      Zahlen auf der Tasta
tur fuer diejeweilige Eingabe.
              Bei der Paperfarbe a
uch; aber     wenn man Ton haben w
ill, muss     man den Wert um 16 e
rhoehen !"
  95 PAUSE 0: PAUSE 0
  98 REM █ Menue █
 100 CLS : PRINT FLASH 1;"
      MENUE                 "
 105 PRINT AT 2,0;"          Borde
r-Effekte:"
 110 PRINT AT 4,0;"1. Blinkende
laufende Streifen  2. Einfarbige
r schnell laufender   Streifen
                   3. Standhafte
 Streifen mit         blinkendem
 Hintergrund       4. Demo
                   5. Blitz-Effe
kt"
 120 PRINT AT 14,0;"Menuepunkt a
uswaehlen !"
 130 IF INKEY$="1" THEN LET u=10
00: GO TO 200
 140 IF INKEY$="2" THEN LET u=12
00: GO TO 300
 150 IF INKEY$="3" THEN LET u=13
00: GO TO 200
 160 IF INKEY$="4" THEN CLS : GO
 TO 1400
 170 IF INKEY$="5" THEN GO TO 40
0
 180 GO TO 130
 190 REM █ Werte werden
           eingegeben █
 200 GO SUB 205
 204 GO TO 225
 205 CLS : PRINT "Werte eingeben
"
 210 INPUT "Dauer (1-255)";s
 220 IF s<1 OR s>255 THEN GO TO
210
 222 PRINT AT 2,0;"Dauer: ";s
 224 RETURN
 225 LET q=1
 230 PRINT AT 20,0;q;". Ink-Farb
e (0-7, -1 fuer keine Farbe)"
 240 INPUT i
 250 IF i<-1 OR i>7 THEN GO TO 2
40
 251 IF q=1 THEN LET a=i: PRINT
AT 4,0;"1. Ink-Farbe: ";a
 252 IF q=2 THEN LET b=i: PRINT
AT 6,0;"2. Ink-Farbe: ";b
 253 IF q=3 THEN LET c=i: PRINT
AT 8,0;"3. Ink-Farbe: ";c
 254 IF q=4 THEN LET d=i: PRINT
AT 10,0;"4. Ink-Farbe: ";d
 255 IF q=4 THEN PRINT AT 20,0;"
     ": GO TO 270
 257 PRINT AT 20,0;"
                        "
 260 LET q=q+1: GO TO 230
 270 INPUT "Paper-Farbe (0-7 ohn
e Ton, 16-23 mit Ton)";e
 280 IF e<0 OR e=8 OR e=9 OR e=1
0 OR e=11 OR e=12 OR e=13 OR e=1
4 OR e=15 OR e>23 THEN GO TO 270
```

Basic-Listing »Border«

```
 290 PRINT AT 12,0;"Paper-Farbe:
 ";e: PAUSE 0: GO TO u
 300 GO SUB 205
 310 INPUT "Ink-Farbe (0-7)";a
 320 IF a<0 OR a>7 THEN GO TO 31
0
 330 PRINT AT 4,0;"Ink-Farbe: ";
a
 340 GO TO 270
 400 GO SUB 205
 440 PAUSE 0: GO TO 1500
1000 REM ■ Blinkende laufende
          Streifen ■
1010 FOR x=0 TO s
1020 IF a>=0 THEN OUT 254,a
1030 IF b>=0 THEN OUT 254,b
1040 IF c>=0 THEN OUT 254,c
1050 IF d>=0 THEN OUT 254,d
1060 OUT 254,e
1070 NEXT x
1080 LET p=1010: LET l=7: LET r=
1090: GO TO 9000
1090 PRINT AT 10,0;"s=";s
1100 PRINT "a=";a: PRINT "b=";b:
 PRINT "c=";c: PRINT "d=";d
1110 PRINT "e=";e
1170 GO TO 9100
1200 REM ■ Einfarbiger schnell
          blinkender laufender
          Streifen ■
1210 FOR x=0 TO s
1220 OUT 254,a: OUT 254,e
1230 NEXT x
1240 LET p=1210: LET l=3: LET r=
1250: GO TO 9000
1250 PRINT AT 5,0;"s=";s: PRINT
"a=";a: PRINT "e=";e
1260 GO TO 9100
1300 REM ■ Standhafte Streifen
          auf blinkenden
          Hintergrund ■
1310 FOR x=0 TO s
1320 PAUSE 1
1330 OUT 254,a: OUT 254,b: OUT 2
54,c: OUT 254,d: OUT 254,e: OUT
254,x
1340 NEXT x
1350 LET p=1310: LET l=6: LET r=
1360: GO TO 9000
1370 GO TO 1090
1400 REM ■ Demo ■
1410 FOR x=200 TO 230
1420 FOR g=190 TO 240
1430 OUT 254,x: OUT 254,g
1440 NEXT g: NEXT x
1450 LET p=1410: LET l=4: LET r=
1460: GO TO 9000
1460 GO TO 9100
1500 REM ■ Blitz-Effekt ■
1510 FOR x=0 TO s
1520 FOR g=0 TO 7
1530 OUT 254,g: OUT 254,g+0
1540 NEXT g: NEXT x
1550 LET p=1510: LET l=4: LET r=
1560: GO TO 9000
1560 PRINT AT 6,0;"s=";s: GO TO
9100
9000 REM ■ Border-Effekt-Listing
          erscheint auf dem
          Bildschirm ■
9010 PRINT AT 15,0;"Soll das Lis
ting zu diesem      Border-Effek
t auf dem Bildschirmerscheinen (
j oder n eingeben)? Nachdem scro
ll? erscheint,       n eingeben u
nd GO TO 9050        entern."
9020 INPUT a$
9030 IF a$="n" OR a$="N" THEN GO
 TO 100
9040 IF a$="j" OR a$="J" THEN CL
S : LIST p
9045 GO TO 9020
9050 FOR f=l TO 21: PRINT AT f,0
;"
   ": NEXT f: GO TO r
9100 REM ■ Border-Effekt-Listing
          wird auf dem Drucker
          ausgegeben ■
9110 PRINT AT 20,0;"COPY (j oder
 n eingeben)?"
9120 INPUT a$
9130 IF a$="j" OR a$="J" THEN PR
INT AT 20,0;"
   ": COPY : GO TO 100
9140 IF a$="n" OR a$="N" THEN GO
 TO 100
9150 GO TO 9120
9900 REM ■ SAVEn des Programms
          Border-Effekte ■
9910 CLS : SAVE "Effekte" LINE 1
9920 PRINT "Band zurueckspulen f
uer 'Verify'"
9930 VERIFY "Effekte"
```

Basic-Listing »Border«

CLEAR-LIST: Mehr Übersicht für Ihr Listing

Bei 32 Zeichen pro Zeile führt schon eine IF/THEN-Anweisung zu einem Zeilenüberlauf. Fängt man erst an, mehrere Befehle in eine Zeile zu packen, muß man bei GO TO- und GOSUB-Befehlen oft lange suchen, bis man die Zeilennummer gefunden hat.

Abhilfe schafft das Programm "Clear-List", in dem es eine linke Spalte (5 Zeichen) für die Zeilennummern bildet und in die rechte Spalte den Zeilentext hineinschreibt. Die Zeilennummern sind dadurch klar in einer Spalte untereinander zu erkennen. Ein Beispiel dafür ist das Programmlisting Clear-List +. Das Maschinenprogramm ist 149 Zeichen lang und so schnell wie das »Original-LIST«. Es kann auf Drucker oder Screen ausgeben, der Ausgabebereich kann durch das Begleitprogramm festgelegt werden.

Verblüffend einfache Eingabe durch Basic-Lader

Das Pogramm geben Sie wie folgt ein: Tippen Sie das Listing »Clear-List« ein. Nun müssen Sie sich überlegen, wo Sie den Code unterbringen. (Zum Beispiel 30000 für den Spectrum 16 KByte). Schützen Sie diesen Bereich mit CLEAR Startadresse — 1 (zum Beispiel 29999). Nun können Sie mit RUN starten. Geben Sie auf Frage die Startadresse an. SAVEN Sie das Programm wie den CODE (mit der Länge 149) ab. Mit NEW löschen Sie das Programm. Geben Sie nun das Begleitprogramm ein. Nach RUN fragt Sie das Programm nach Ausgabegerät und Bereich. Mit 2,1,10000 wird zum Beispiel das Begleitprogramm gelistet. Treten Fehler auf, muß das Eingabeprogramm überprüft werden! Nachdem Sie nun das Maschinencodeprogramm Ihren Wünschen entsprechend angepaßt haben, können Sie mit Load "" ein Programm nach dem anderen einladen und mit RANDOMIZE USR Startadresse nach Herzenslust listen lassen.

(Hans Georg v. Zeschwitz)

F000—F004	Kanal (Screen (2)/Printer (3) öffnen)
F005—F00A	1. Zeilennr. suchen
F00B—F010	letzte Zeilennr. ? (ja: Rücksprung in Basic)
F012	Zeilennr. ausdrucken
F015—F019	Tabulator in c auf 5 setzen, ein Leerzeichen drucken.
F01A—F01D	Hl auf 1. Zeichen setzen, in a laden
F01E—F024	Enter ? (Neue Zeile, hl erhöhen und Sprung wenn ja)
F026—F02E	Versteckte Zahl? (Hl +6 und Sprung, wenn ja)
F030—F038	Token ? (Ausdrucken mit Routine und nächstes Zeichen, bei nein)
F03A—F050	Wenn a ein Leerzeichen vorher benötigt (z.B. OR) dieses ausdrucken.
F052	a mit der Nr. des Token beschreiben (RND erstes)
F054—5a	Hl und bc retten, nach bc und hl die Startadresse im ROM laden, wo die Zeichen der Tokens stehen (durch 7. Bit wird dabei das Ende signalisiert).
F05B—F061	Soviele Tokenenden zählen, wie in a stehen. bc enthält die Anfangsadresse des aktuellen Token.
F063—F065	Nach Hl Tokenanfang, bc Original
F066—F069	ein Zeichen laden, Bit 7 löschen und ausdrucken.
F06C—F06F	so lange weitermachen, bis man ans Ende (Bit 7) kommt.
F071—F07d	Register (außer de) original, wenn Leerzeichen hinter Token erforderlich, ausdrucken, dann nächstes Zeichen laden
F080—F094	Diese Ausdrucksroutine ist der Kern des Programms, sie ist lediglich dazu da, falls das Ende einer Zeile erreicht ist, 5 Leerzeichen auszudrucken und c neu zu setzen.

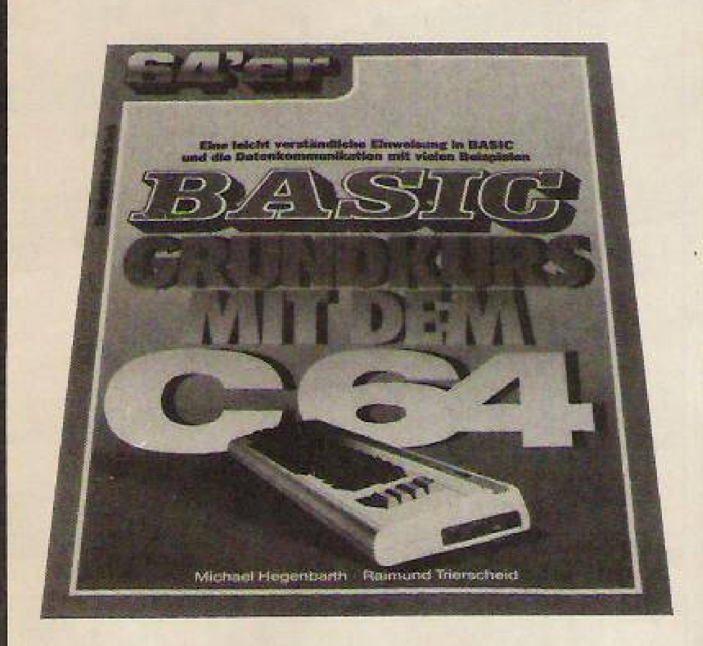
Der Aufbau des Programmes

```
  10 REM Clear-List
  20 DEF FN d(a$)=CODE a$-48-(7
AND a$(1)>"9")
  30 DATA "3E03CD0116210100CD6E1
9011027CD8019D0CD281A2323237EC"
  40 DATA "DB61823FE0D20030718E7
FEA53005CDPPPP18EBD6A5119500F5"
  50 DATA "CD410C3E20D4PPPP1AE67
FCDPPPP1A138730F5F1FE033E20D4PP"
  60 DATA "PP18C8F5D5E5CD03087DE
61F20053E05CDC30AE1D1F147D778C9"
  70 CLEAR USR "A"-110: LET z=US
R "a"-108: LET c=z+80
  80 LET a$="": FOR n=1 TO 4: RE
AD b$: LET a$=a$+b$: NEXT n
  90 LET s=0: FOR n=z TO z+102
 100 IF a$(1)<>"P" THEN GO TO 13
0
 110 LET a$=a$(5 TO ): POKE n,c-
INT (c/256)*256
 120 LET n=n+1: POKE n,INT (c/25
6): NEXT n
 130 LET m=FN d(a$)*16+FN d(a$(2
)): LET a$=a$(3 TO )
 140 POKE n,m: LET s=s+m: NEXT n
 150 IF s<>10500 THEN PRINT "Err
or": STOP
 160 PRINT "Alles ok"'"Aufruf mi
t RANDOMIZE USR ";z
 170 PRINT "POKE ";z+1;",2 druck
t auf Screen"
 180 PRINT "POKE ";z+1;",3 druck
t auf Printer"
 190 PRINT "SAVEN mit"'"SAVE ""C
L""CODE ";z;",105"
 200 PRINT "Zum spaeteren Laden:
"'"CLEAR ";z-1'"LOAD ""CL""CODE
"
 210 COPY : PAUSE 0: CLS
```

Eingabe-Programm für »Clear-List«

Markt&Technik-Buchverlag

Willkommen bei Commodore 64



M. Hegenbarth/R. Trierscheid

BASIC-Grundkurs mit dem Commodore 64
März 1985, 377 Seiten
Eine grundlegende, leicht verständliche Einführung in die BASIC-Programmierung des Commodore 64 · die ersten »Gehversuche« · ein Programm schreiben · alles über Variablen · die Universalanweisung PRINT · Sprünge und Schleifen · die Eingabe von Daten · zwei C64 kommunizieren miteinander: Erläuterung einfach gehaltener BASIC-Programme, die die Datenfernübertragung per Akkustikkoppler steuern · Übersicht aller BASIC-Befehle mit Syntax und Erläuterungen im Anhang · ein Buch, das durch seinen praxisbezogenen Aufbau einen mühelosen Einstieg in die BASIC-Programmierung mit dem C64 ermöglicht!
Best.-Nr. MT 633
(Sfr. 40,50/öS 343,20) **DM 44,–**



G. Beekman

Ihr Heimcomputer Commodore 64
August 1984, 296 Seiten
Alles Wissenswerte im Umgang mit dem Commodore 64 · Planung, Kauf und Inbetriebnahme der Anlage · Einsatz fertig gekaufter oder selbst erstellter Programme · Schwächen und Stärken der altbewährten und neuesten Programmiersprachen · die gängigsten Software-Angebote für jeden Einsteiger.
Best.-Nr. MT 701
(Sfr. 35,–/öS 296,40) **DM 38,–**



E. H. Carlson

Basic mit dem Commodore 64
April 1984, 320 Seiten
Ein Basic-Lehrbuch für den jugendlichen Anfänger · übersichtlich gegliederte Lernprogramme · Alles über INPUT-GOTO · Let-Befehle · Editorfunktionen · POKE-Befehle für die Grafik · geeignet auch als Leitfaden für Lehrer und Eltern.
Best.-Nr. MT 657
(Sfr. 44,20/öS 374,40) **DM 48,–**



J. W. Willis/D. Willis

Commodore 64 — leicht verständlich
Juni 1984, 154 Seiten
Informationen für den Computer-Neuling · Installation und Inbetriebnahme · Programmieren in Basic · Grafik und Töne · Auswahl von Hardware und Zubehör · Software für Ihren Computer · die ideale Einführung in das Arbeiten mit Ihrem Commodore 64.
Best.-Nr. MT 700
(Sfr. 27,50/öS 232,40) **DM 29,80**



T. Rugg/Ph. Feldman

Mehr als 32 Basic-Programme für den Commodore 64
März 1984, 279 Seiten
Programme speziell für den Commodore 64 · umfassende praktische Anwendungen · jede Menge Lehr- und Lernhilfen · super Spiele · für Basic-Neulinge und Experten.
Best.-Nr. MT 613 (Buch)
(Sfr. 45,10/öS 382,20) **DM 49,–**
Best.-Nr. MT 614 (Beispiele auf Diskette)
(Sfr. 48,–/öS 432,–) ***DM 48,–**
* inkl. MwSt. Unverbindliche Preisempfehlung

Lehrspielzeug Computer: C 64/VC-20. **Juli 1984, 120 Seiten**
Speziell für Kinder entwickelt führt dieses Buch spielerisch in die Basic-Welt des C 64/VC-20 ein · mit vielen lehrreichen Spielprogrammen und Grafikmöglichkeiten · kleinere Kinder benötigen die Hilfe ihrer sachkundigen Eltern.
Best.-Nr. MT 695 DM 24,80 (Sfr. 23,–/öS 193,40)

Das große Spielebuch — Commodore 64 **Februar 1984, 141 Seiten**
46 Spielprogramme · Wissenswertes über Programmiertechnik · praxisnahe Hinweise zur Grafikherstellung · alles über Joystick- und Paddleansteuerung · das Spielebuch mit Lerneffekt.
Best.-Nr. MT 603 (Buch) DM 29,80 (Sfr. 27,50/öS 232,40)
Best.-Nr. MT 604 (Beispiele auf Diskette) *DM 38,– (Sfr. 38,–/öS 342,–)
* inkl. MwSt. Unverbindliche Preisempfehlung

Einführungskurs: Commodore 64 **Mai 1984, 276 Seiten**
Die Programmiersprache Basic · Einsatzgebiete des Commodore 64-Basic: Grafik, Musik, Dateiverwaltung · mit vielen Beispielprogrammen, häufig benötigten Tabellen und nützlichen Tips · für Einsteiger und Fortgeschrittene.
Best.-Nr. MT 685 DM 38,– (Sfr. 35,–/öS 296,40)

Computer für Kinder Ausgabe Commodore 64 **1984, 112 Seiten**
Ein Buch für Kinder und ihre Lehrer · ideal für die erste Begegnung mit Computern, ihren Eigenwilligkeiten und ihren unerschöpflichen Möglichkeiten · leichtverständliche Erläuterungen rund um den Commodore 64 · alle Programmbeispiele in BASIC.
Best.-Nr. PW 709 DM 29,80 (Sfr. 27,50/öS 232,40)

Commodore 64 Listings — Band 1: Spiele **Oktober 1984, 199 Seiten**
Mit ausführlicher Dokumentation · Spielanleitung · Variablen für die Änderung der Spiele · vollständige Listings für: Bürger Joe · Nibbler · Zingel Zangel · Universe · Würfelpoker · Maze-Mission · der magische Kreis · Todeskommando Atlantik · Enterprise.
Best.-Nr. MT 748 DM 24,80 (Sfr. 23,–/öS 193,40)
Best.-Nr. MT 804 (Beispiele auf Diskette) *DM 38,– (Sfr. 38,–/öS 342,–)
* inkl. MwSt. Unverbindliche Preisempfehlung

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Fragen Sie dort nach unserem Gesamtkatalog mit über 170 neuen Computerbüchern.



Bestellkarten bitte an Ihren Buchhändler oder an einen unserer Depot-Händler. Adressenverzeichnis am Ende des Heftes. Beim Markt & Technik Verlag eingehende Bestellungen werden von den Depot-Händlern ausgeliefert.

Markt & Technik Verlag AG Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik-Vertriebs AG, Kollerstraße 3, CH-6300 Zug, ☎ 042/223155
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien ☎ 0222/677526

Die angegebenen Preise sind Ladenpreise.

```
F000  ld a,03           3E 03
F002  call 1601         CD 01 16
F005  ld hl,0001        21 01 00
F008  call 196E         CD 6E 19
F00B  ld bc,2710        01 10 27
F00E  call 1980         CD 80 19
F011  ret nc            D0
F012  call 1A28         CD 28 1A
F015  inc hl            23
F016  inc hl            23
F017  inc hl            23
F018  ld a,(hl)         7E
F019  call 18B6         CD B6 18
F01C  inc hl            23
F01D  cp 0D             FE 0D
F01F  jr nz,>F024       20 03
F021  rst 16            D7
F022  jr >F00B          18 E7
F024  cp EA             FE EA
F026  jr z,>F02C        28 04
F028  cp F1             FE F1
F02A  jr nz,>F02E       20 02
F02C  ld a,20           3E 20
F02E  cp A5             FE A5
F030  jr nc,>F037       30 05
F032  call F05A         CD 5A F0
F035  jr >F018          18 E1
F037  sub A5            D6 A5
F039  ld de,0095        11 95 00
F03C  push af           F5
F03D  call 0C41         CD 41 0C
F040  ld a,20           3E 20
F042  call nc,F05A      D4 5A F0
F045  ld a,(de)         1A
F046  and 7F            E6 7F
F048  call F05A         CD 5A F0
F04B  ld a,(de)         1A
F04C  inc de            13
F04D  add a,a           87
F04E  jr nc,>F045       30 F5
F050  pop af            F1
F051  cp 03             FE 03
F053  ld a,20           3E 20
F055  call nc,F05A      D4 5A F0
F058  jr >F018          18 BE
F05A  push af           F5
F05B  push de           D5
F05C  push hl           E5
F05D  call 0B03         CD 03 0B
F060  ld a,l            7D
F061  and 1F            E6 1F
F063  jr nz,>F06A       20 05
F065  ld a,05           3E 05
F067  call 0AC3         CD C3 0A
F06A  pop hl            E1
F06B  pop de            D1
F06C  pop af            F1
F06D  ld b,a            47
F06E  rst 16            D7
F06F  ld a,b            78
F070  ret               C9
```

Maschinencode für 64-KByte-Bereich

```
E000  ld a,03           3E 03
E002  call 1601         CD 01 16
E005  ld hl,0001        21 01 00
E008  call 196E         CD 6E 19
E00B  ld bc,2710        01 10 27
E00E  call 1980         CD 80 19
E011  ret nc            D0
E012  call 1A28         CD 28 1A
E015  inc hl            23
E016  inc hl            23
E017  inc hl            23
E018  ld a,(hl)         7E
E019  call 18B6         CD B6 18
E01C  inc hl            23
E01D  cp 0D             FE 0D
E01F  jr nz,>E024       20 03
E021  rst 16            D7
E022  jr >E00B          18 E7
E024  cp A5             FE A5
E026  jr nc,>E02D       30 05
E028  call E050         CD 50 E0
E02B  jr >E018          18 EB
E02D  sub A5            D6 A5
E02F  ld de,0095        11 95 00
E032  push af           F5
E033  call 0C41         CD 41 0C
E036  ld a,20           3E 20
E038  call nc,E050      D4 50 E0
E03B  ld a,(de)         1A
E03C  and 7F            E6 7F
E03E  call E050         CD 50 E0
E041  ld a,(de)         1A
E042  inc de            13
E043  add a,a           87
E044  jr nc,>E03B       30 F5
E046  pop af            F1
E047  cp 03             FE 03
E049  ld a,20           3E 20
E04B  call nc,E050      D4 50 E0
E04E  jr >E018          18 C8
E050  push af           F5
E051  push de           D5
E052  push hl           E5
E053  call 0B03         CD 03 0B
E056  ld a,l            7D
E057  and 1F            E6 1F
E059  jr nz,>E060       20 05
E05B  ld a,05           3E 05
E05D  call 0AC3         CD C3 0A
E060  pop hl            E1
E061  pop de            D1
E062  pop af            F1
E063  ld b,a            47
E064  rst 16            D7
E065  ld a,b            78
E066  ret               C9
```

Maschinencode für 32-KByte-Bereich

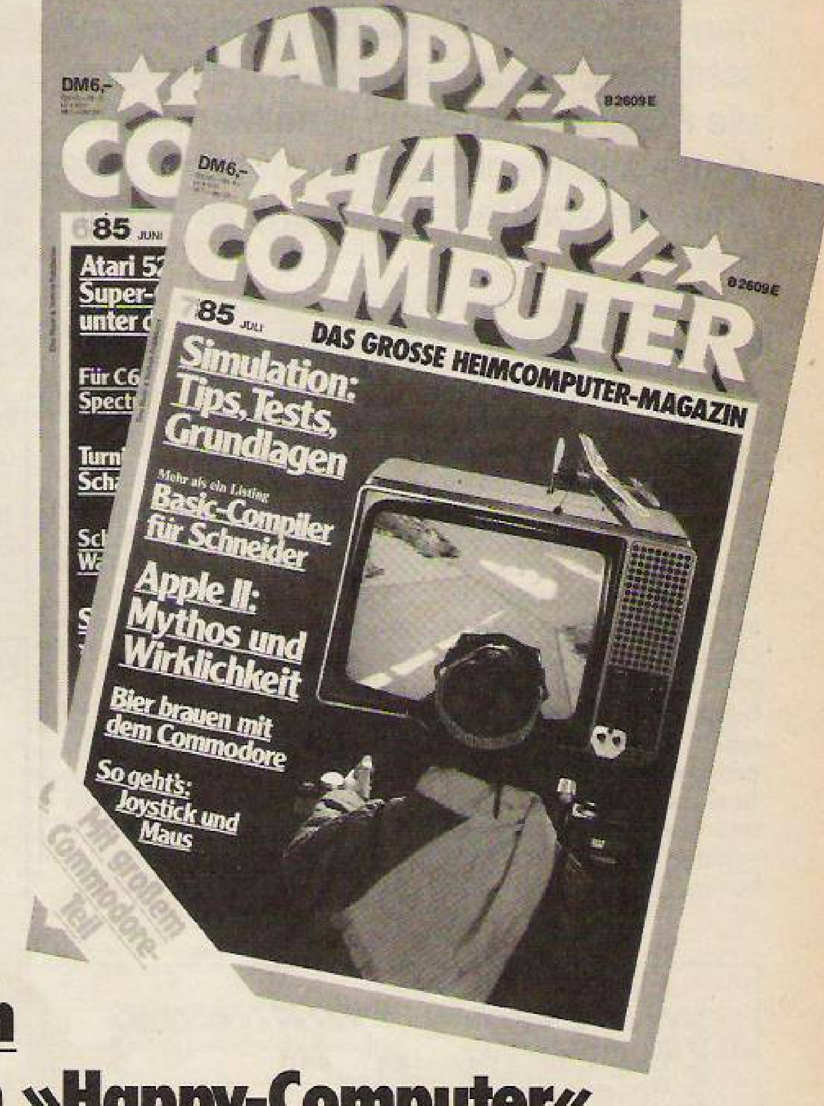
```
 10 Clear-List +
 20 DEF FN d(a$)=CODE a$-48-(7
    AND a$(1)>"9")
 30 DATA "3E03CD0116210100CD6E1
    9011027CD8019D0CD281A232323
    7ECDB618"
 40 DATA "23FE0D2003D718E7FEEA2
    804FEF120023E20FEA53005CDPP
    PP18E1D6"
 50 DATA "A5119500F5CD410C3E20D
    4PPPP1AE67FCDPPPP1A138730F5
    F1FE033E2"
 60 DATA "0D4PPPP18BEF5D5E5CD03
    0B7DE61F20053E05CDC30AE1D1F
    147D778C9"
 70 CLEAR USR "A"-120: z=USR "a
    "-118: c=z+90
 80 a$="": FOR n=1 TO 4: READ b
    $: a$=a$+b$: NEXT n
 90 s=0: FOR n=z TO z+112
100 IF a$(1)<> "P" THEN  GO TO
    130
110 a$=a$(5 TO ): POKE n,c-INT
    (c/256)*256
120 n=n+1: POKE n,INT (c/256):
    NEXT n
130 m=FN d(a$)*16+FN d(a$(2)):
    a$=a$(3 TO )
140 POKE n,m: s=s+m: NEXT n
150 IF s<> 11635 THEN  PRINT "E
    rror": STOP
160 PRINT "Alles ok"'"Aufruf mi
    t RANDOMIZE USR ";z
170 PRINT "POKE ";z+1;",2 druck
    t auf Screen"
180 PRINT "POKE ";z+1;",3 druck
    t auf Printer"
190 PRINT "SAVEN mit"'"SAVE ""C
    L""CODE ";z;",115"
200 PRINT "Zum spaeteren Laden:
    "'"CLEAR ";z-1'"LOAD ""CL""
    CODE "
210 COPY : PAUSE 0: CLS
```

Eingabe-Programm für »Clear-List +«

Entdecken Sie jetzt mit »Happy-Computer« die ganze Welt der Heimcomputer!

Denn »Happy-Computer« ist einfach super:

»Happy-Computer« bringt jeden Monat neue, tolle Tips und interessante Informationen über alle wichtigen Homecomputer-Systeme. In »Happy-Computer« finden Sie Monat für Monat Listings der schönsten Homecomputer-Programme, die unsere Redakteure für Sie geprüft haben, die neuesten Hard- und Software-Tests sowie pfiffige Programmier- und Anwendungsbeispiele. All das und vieles mehr macht »Happy-Computer« zu einer wichtigen und interessanten Fachzeitschrift für alle, die Spaß an Homecomputern haben.



Nutzen Sie deshalb jetzt die Vorteile durch ein Abonnement von »Happy-Computer«

Fünf gute Gründe sprechen dafür:

1. »Happy-Computer« erscheint bereits Mitte des Vormonats: So können Sie brandneue Listings bereits früher als sonst in Ihren Computer eingeben, Testberichte lesen und schneller auf interessante Angebote reagieren.

2. »Happy-Computer« kommt jeden Monat, pünktlich und bequem, direkt ins Haus.

3. Sie sind lückenlos informiert: Als Abonnent können Sie sicher sein, daß Sie Monat für Monat, Ausgabe für Ausgabe, »Happy-Computer« pünktlich erhalten – auch wenn Sie im Urlaub sind oder das Heft beim Händler vergriffen ist. Es gehen Ihnen also keine Informationen verloren.

4. Nur als Abonnent erhalten Sie 12 Hefte zum Preis von 11: Sie bezahlen (im Inland) nur DM 66,– für ein Jahr statt DM 72,– im Einzelverkauf. Das sind fast 10% Preisvorteil!

5. Es entstehen Ihnen keine weiteren Kosten: Porto und Zustellgebühren übernimmt der Verlag.

Gutschein

FÜR EIN KOSTENLOSES PROBEEXEMPLAR VON HAPPY COMPUTER

JA, ich möchte »Happy-Computer« kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »Happy-Computer« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »Happy-Computer« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur DM 66,— statt DM 72,— Einzelverkaufspreis (Ausland auf Anfrage).

Vorname, Name

Straße | PLZ, Ort

Datum | 1. Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Datum | 2. Unterschrift

Gutschein ausfüllen, ausschneiden, in ein Kuvert stecken und absenden an: Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar

Bit-Kopierer

Wer schon einmal versucht hat, sich eine Sicherheitskopie von einem gekauften Programm zuzulegen, um das Original vor einer Zerstörung durch einen defekten Kassettenrecorder zu schützen, wird feststellen, daß sich die Programmautoren eine Menge Gedanken gemacht haben, um ihre Programme vor Raubkopierern zu schützen.

Wer aber trotzdem nicht auf eine Sicherheitskopie verzichten will, kann sich freuen. Das einzige was man braucht, ist ein zweiter Kassettenrecorder und dieses Programm. Wer diese Zeilen liest wird wahrscheinlich denken, daß, wenn man einen zweiten Kassettenrecorder benutzt, gleich die beiden Recorder miteinander verbinden und dann direkt überspielen kann. Dies mag in einigen Fällen stimmen, doch oft ist es so, daß der dadurch entstehende Qualitätsverlust die Ladefähigkeit des Spectrum übersteigt. Wenn man ein Programm lädt und danach wieder abspeichert, erhält man eine Kopie von hoher Qualität. Mein Gedanke war, das Laden und Speichern gleichzeitig machen zu lassen und damit eine Kopie von gleich hoher Qualität zu erhalten. Der Spectrum wäre damit ein Aufbereiter oder Verstärker des Signales. Damit das Einlesen und Ausgeben mit extrem hoher Geschwindigkeit ging, schrieb ich eine nur 34 Byte kurze Assemblerroutine. Ich rechnete mir aus, daß das Einlesen und Ausgeben des Signals schneller gehen mußte, als die SAVE- oder LOAD-Routine des Spectrums. Ein weiterer Vorteil ist, daß mit dem Programm kein Byte an Speicher verloren geht, was auch das Kopieren von Programmen möglich macht, die den Spectrum voll ausnutzen. Der Abspielrecorder wird an den EAR-Eingang angeschlossen und der Aufnahmerecorder an den MIC-Ausgang. Die beiden Recorder werden nun genau so eingestellt, wie es beim normalen SAVE- oder LOAD-Vorgang getan wird. Wird nun das Programm gestartet, braucht nur noch das getan zu werden, was auf dem Bildschirm gezeigt wird. Da die Assemblerroutine nicht feststellen kann, wann ein Überspielvorgang beendet ist, muß als Abschluß »SPACE« gedrückt werden.

Zum Ablauf

Ist das Programm abgetippt, so kann es mit »RUN 100« gesichert werden. Nach dem folgenden VERIFY wird sich zeigen, ob das Programm richtig eingegeben wurde. Erscheint »Fehler in der Data-Liste« so bedarf dies einer Kontrolle der Prüfsumme in Zeile 270 (IF s< >5239...) und der DATA-Liste in Zeile 300. Findet man den Fehler nicht, so läßt sich oft durch die falsche Prüfsumme, die in Speicher s steht, ein Rückschluß ziehen.

Am Schluß muß noch bemerkt werden, daß mit dem Programm ausschließlich Sicherheitskopien für den Privatgebrauch gemacht werden dürfen. (Uwe Roth)

Programmaufbau:

003 — 070	:	Hauptprogramm. Erzeugen des Bildschirms und Aufruf der Maschinenroutine.
100 — 120	:	Abspeichern des Programms mit Verify.
200 — 290	:	Erzeugen der Maschinenroutine.
300 — 310	:	Maschinenroutine als DATA-Liste.

Variablenliste:

s	:	Prüfsumme
i	:	Läufervariable beinhaltet die Adresse des momentan abzuspeichernden Bytes zur Erzeugung des Maschinenprogramms.
x	:	Maschinenbefehl als Zahl. Sie wird aus dem DATA-Feld gelesen und an die Adresse i gepoket.

```
   1 REM
 Bit-Copierer fuer den
 ZX Spectrum 16/48K

 von Uwe Roth
 © by Real Roth MCMLXXXIV

 Adresse:
 Uwe Roth
 Irscherstrasse 15
 5500 Trier
 Tel.:0651/17774
   3 REM
Hauptprogramm
   5 BORDER 1: PAPER 6: CLEAR 29
999
  10 GO SUB 200
  20 CLS : PRINT AT 2,0;"Bit-Cop
ierer"''"von Uwe Roth"''"© by Real
 Roth"''"MCMLXXXIV";AT 9,0;"Kasse
ttenrekorder 1:Aufnahme"''"Kasse
ttenrekorder 2:Band starten"
  30 PRINT ''"Space beendet Uebe
rspielvorgang"
  40 RANDOMIZE USR 30000
  50 BORDER 1: CLS : PRINT AT 10
,10;"Das wars..."
  60 STOP : GO TO 20
  70 REM
Programm absaven
 100 SAVE "File COPY " LINE 0: P
RINT " VERIFY ": VERIFY ""
 110 GO TO 5
 120 REM
Maschinenroutiene erzeugen
 200 RESTORE 300
 210 LET s=0
 220 FOR i=30000 TO 30034
 230 READ x
 240 LET s=s+x
 250 POKE i,x
 260 NEXT i
 270 IF s<>5239 THEN PRINT AT 10
,0;"Fehler in der Data-Liste": S
TOP
 280 RETURN
 290 REM
Maschinenroutiene als DATA -Feld
 300 DATA  243,  1,254,127,219,
254,203,119,202, 70,117, 62, 13,
211,254, 62,  5,211,254,195, 74,
117, 62,  2,211,254,237,120,203,
 71,194, 49,117,251,201
 310 REM
          ** Ende **
```

Basic-Listing »Bit-Kopierer«

Aufbau der Maschinenroutine:

000	Festsetzung des ersten Befehls auf die Adresse 30000.
010	Sperren des Interrupts.
020	Laden der Portadresse 32766 (Tastenreihe von SPACE bis B) in BC
030	Einlesen des Ports 254 (Bit 6 ist EAR-Eingang)
040 — 050	Testen des 6. Bits auf 0 oder 1 Wenn 0: Sprung zu »NULL« Wenn 1: Weiter zu »EINS«
060 — 090	Ausgabe an Port 254 (Bit 4 = Lautsprecher; Bit 3 = MIC-Ausgang; Bit 0-2 = Border Farbe) mit 0 für Bit 4, 1 für Bit 3 und 5 (entspricht cyan) für Bit 0-2. Daraufhin folgt dieselbe Ausgabe mit Bit 3 = 1.
100	Sprung zu »CONT«
110 — 120	Ausgabe an Port 254 mit 0 für Bit 3 und 4 und mit 2 (entspricht Rot) für Bit 0-2. Durch setzen der Farben Cyan für 0 am EAR-Eingang und Rot für 1 am EAR-Eingang entstehen dieselben charakteristischen Streifen wie beim Laden eines normalen Programms. Treten sie nicht auf, so ist das Eingangssignal zu schwach.
130 — 150	Abfrage ob »SPACE« gedrückt wurde durch Port BC (32766), Bit 0. Wenn nicht gedrückt wurde: Sprung zu »LOOP«.
160	Öffnen des Interrupts.
170	Rücksprung ins BASIC

```
7530          0000         ORG  30000
7530 F3       0010 BCOPY   DI
7531 01FE7F   0020 LOOP    LD   BC,32766
7534 DBFE     0030         IN   A,(254)
7536 CB77     0040         BIT  6,A
7538 CA4675   0050         JP   Z,NULL
753B 3E0D     0060 EINS    LD   A,13
753D D3FE     0070         OUT  (254),A
753F 3E05     0080         LD   A,5
7541 D3FE     0090         OUT  (254),A
7543 C34A75   0100         JP   CONT
7546 3E02     0110 NULL    LD   A,2
7548 D3FE     0120         OUT  (254),A
754A ED78     0130 CONT    IN   A,(C)
754C CB47     0140         BIT  0,A
754E C23175   0150         JP   NZ,LOOP
7551 FB       0160         EI
7552 C9       0170         RET
              0180         END

CONT    754A
NULL    7546
EINS    753B
LOOP    7531
BCOPY   7530
#       5DCE
```

Maschinencode-Listing »Bit-Kopierer«

UDGs durch POKE-Befehle

Mit diesem Hilfsprogramm können Sie Grafik-Zeichen auf dem Bildschirm darstellen, ohne diese vorher zu definieren.

Man kann zum Beispiel mit Hilfe dieses Programms Muster auswählen, um diese in eigene Spiele als Hintergrund oder als Mauern einzubauen.

Wenn das Programm eingegeben und gestartet worden ist, muß man eine Zahl zwischen 0 und 255 eingeben (viele Variationsmöglichkeiten).

Nun bekommt man verschiedene Muster gezeigt. (Beliebige Taste drücken, damit neues Muster erscheint; Q (mit Caps Shift) drücken, damit man neue Variationszahl eingeben kann.) Unten steht jeweils, welches Zeichen man eingeben muß, um dieses Zeichen zu erhalten. Der POKE-Befehl, den man vorher eingeben muß, erscheint auch auf dem Bildschirm.

Der Zeichensatz wird durch den POKE-Befehl verändert. Um den Zeichensatz wieder normal darstellen zu können, muß man POKE 23607,60 eingeben. Mit GO TO 300 kann man das Programm »UDGs durch POKE-Befehle« auf Band sichern:

Aufschlüsselung nach Zeilennummern:

120 — 130 : Eingabe der Pokezahl
140 — 210 : Hauptprogramm:
145 : Zeichensatz wird durch Pokezahl verändert
150 — 180 : Muster erscheint auf Bildschirm
190 : Angaben zum Muster erscheinen auf dem Bildschirm
200 : Tastaturabfrage
300 — 320 : SAVEn für Programm: »UDGs durch POKE-Befehle«

Variablenliste:

z POKE-Wert, um den Zeichensatz zu verändern
f Schleifenvariable, damit alle Zeichen des Zeichensatz erscheinen
g,h Schleifenvariablen zum Zeichnen des Musters

(Bernhard Baran)

```
 100 REM © Bernhard Baran
           Postfach 35
           6710 Frankenthal
           Tel.:06233/24243
 110 REM ■ UDGs durch
           POKE-Befehle ■
 120 INPUT "Eine Zahl zwischen 0
 und 255    eingeben. (60=normal
)";z
 130 IF z<0 OR z>255 THEN GO TO
120
 140 FOR f=32 TO 127
 145 POKE 23607,z
 150 FOR g=0 TO 4
 160 FOR h=0 TO 31
 170 PRINT AT g,h;CHR$ f;AT 6,10
 CHR$ f;AT 6,22;CHR$ f
 180 NEXT h: NEXT g
 190 POKE 23607,60: PRINT #0;AT
0,0;"POKE 23607,";z;AT 1,0;"Zeic
hen:";CHR$ f
 200 PAUSE 0: IF INKEY$="Q" THEN
 RUN
 210 CLS : NEXT f: RUN
 300 CLS : SAVE "UDG" LINE 1
 310 PRINT "Band zurueckspulen f
uer 'Verify'"
 320 VERIFY "UDG"
```

Basic-Listing »UDG«

Funktionstasten trotz Interface 1

In Happy-Computer Ausgabe 1/1985 stand ein sehr schönes Programm von R. Fuchs. Man kann damit jede Spectrumtaste mit einer Funktion belegen. Leider läuft das Programm nur auf einem 48-KByte-Spectrum ohne Interface 1. Ich habe versucht das Programm so umzuschreiben, daß dieser es sowohl mit Interface 1 als auch auf dem 16-KByte-Spectrum läuft.

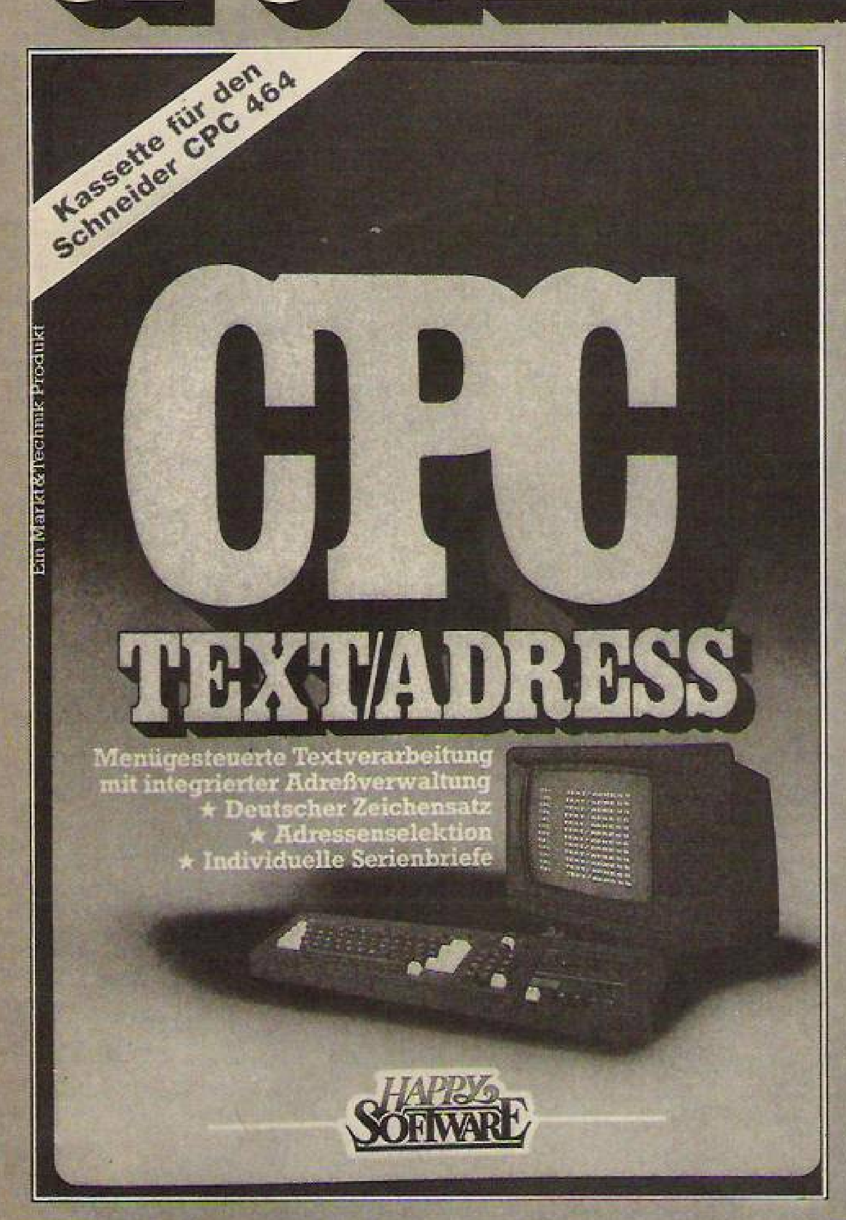
Bei dieser Arbeit hat sich herausgestellt, daß das Programm auf einem 16-KByte-Spectrum nur mit einem schweren Nachteil läuft: einem schlechten Bild. Offensichtlich greift die Interruptmodus Änderung in die Bildsynchronisation ein. Da beim 16-KByte-Spectrum die Funktionstasten Routine in dem gleichen Speicherblock liegt wie der Videospeicher, kommt es zu Kollisionen zwischen beiden. Schade für die 16-KByte-Besitzer.

Warum lief das Programm mit Interface 1 nicht? Der Autor benutzte das ROM für die Adresse der Interrupt-Routine. Dazu überlegen wir uns, was im Interruptmodus 2 passiert. Der Z80 setzt eine Adresse aus dem I-Register und dem Wert auf dem Datenbus (den liefert normalerweise das Gerät, das den Interrupt produziert hat) zusammen. Beim Spectrum liegt auf dem

```
32260   3E3EED47ED56C900  -> 956
32268   00003E7DED4721FF  -> 783
32276   7D362823367EED5E  -> 765
32284   2A535C220B7E3E00  -> 450
32292   320D7EC9FFF3DDE5  -> 1338
32300   E5D5F53A0D7EFE01  -> 1139
32308   283BFDCB016E2861  -> 803
32316   CD1C7EDD2A0B7EDD  -> 980
32324   7E04FEEA2050DD7E  -> 1077
32332   0521085CBE2814DD  -> 609
32340   6E02DD6603ED5B0B  -> 777
32348   7E1911040019220B  -> 242
32356   7E18D8CD877E3E01  -> 895
32364   320D7E182CDD2A0B  -> 531
32372   7EDD7E07FE0D281E  -> 817
32380   FE4020023E0DCD8A  -> 770
32388   7E1816DD7E073208  -> 584
32396   5C213B5CCBEE2A0B  -> 770
32404   7E232206B7EC9CD1C -> 766
32412   7EF1D1E1DDE1FBC9  -> 1699
```

Hex-Listing für 16-KByte-Version

CPC-TEXT/ADRESS



Jetzt neu!

Serienbriefe — kein Problem!
Textverarbeitung und Adreß-verwaltung — ein kombiniertes Paket!

Das Programm unterstützt das Diskettenlaufwerk (Wahlmöglichkeit: Speicherung Ihrer Briefe und Adressen auf Kassette oder Diskette).

Leistungsbeschreibung von CPC-Text:

- Menügesteuerte Bedienerführung
- Automatische Trennvorschläge
- Blocksatz; Tabulatorfunktionen; Blockoperationen
- Deutsche Tastaturanpassung; deutscher Zeichensatz
- Texteingabe im 80-Zeichen-Modus (variable Zeilenbreite)
- Eigener Funktionsteil zur Druckeranpassung
- Cursororientierter Texteditor zur problemlosen Korrektur
- Serienbrieferstellung mit individuellen Empfängeradressen und persönlicher Briefanrede
- Ansteuerung von Drucksonderfunktionen

Leistungsbeschreibung von CPC-Adreß:

- Feste Eingabemaske mit sieben Eingabefeldern
- Ausgabe der selektierten Adressen in eine separate Textdatei
- Auswahlmöglichkeit der Suchroutinen nach Code, Name oder Maske
- Druck auf Endlospapier oder Adreßetiketten

Minimale Hardwareanforderungen:

- Schneider CPC 464
- Beliebiger Drucker mit Centronics-Schnittstelle (standardmäßige Anpassung an alle Epson-Drucker und Schneider NLQ 401)

M&T-Programme:

Ihre ganz persönlichen Problemlösungen

DM 79,— * auf Kassette, Best.-Nr. MK 242 G (ISBN 3-89090-101-8)
(Sfr. 73,—/öS 711,—)

DM 89,— * auf Diskette, Best.-Nr. MD 244 G
(Sfr. 83,—/öS 801,—)

* inkl. MwSt. unverbindliche Preisempfehlung

Als Ergänzung empfehlen wir Ihnen unser neues Buch für den Schneider CPC:

CPC 464 FÜR EIN- UND UMSTEIGER

Carsten Straush
Hartmut Pick



Dieses Buch ist eine praxisorientierte Spiel- und Arbeitshilfe für den Schneider CPC 464.

In einer Rundreise durch die Bereiche BASIC, Grafik, Sound, Tastaturanwendung und Kassettenrecordereinsatz, werden die meisten Befehle des CPC in kompakter, systematischer Form dargestellt. Schwerpunkte sind dabei die im Standard-BASIC nicht enthaltenen Kommandos und ihre Anwendung in einer Reihe nützlicher Programme zur Textverarbeitung, Datenverwaltung, in der Fehlerbehandlung, bei der grafischen Darstellung und im Musikbereich. Die weitgehend modular aufgebauten Beispielprogramme bilden den Grundstock für eine CPC 464-Programmbibliothek.
Das Buch ist für Anfänger und Fortgeschrittene, die sich die Möglichkeiten des Schneider CPC 464 über das Handbuch hinaus erschließen wollen, gleichermaßen gut geeignet.

Best.-Nr. MT 801, ISBN 3-89090-090-9
DM 46,— (Sfr. 44,20/öS 358,80)

Die angegebenen Preise sind Ladenpreise.

Markt & Technik-Produkte erhalten Sie bei Ihrem Buchhändler.

Bestellkarten bitte an Ihren Buchhändler oder an eine unserer Depotbuchhandlungen. Adressenverzeichnis am Ende des Heftes! Beim Markt & Technik-

Markt&Technik
Verlag Aktiengesellschaft
Buchverlag

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik-Vertriebs AG, Kollerstraße 3, CH-6300 Zug, ☎ 042/22 31 55

Datenbus, beim Interrupt immer 255. Das I-Register kann man beliebig laden. Bei der Adresse 256*I+255 schaut der Z80 dann nach der Adresse, an der die Interruptroutine beginnt. Der Autor hatte I=9 gesetzt. Das liefert eine Adresse im ROM (also einen vorgegeben Wert). Und wenn gerade das neue ROM eingeschaltet war ging nichts mehr, da dann natürlich etwas anderes im ROM als Adresse gelesen wurde. Ich habe die Variable MIADD (siehe Disassembler-Listing) im RAM eingeführt, und schreibe die Adresse dort hinein. Dazu habe ich I mit 253 beziehungsweise 125 geladen.

So arbeitet die Routine

Das Laden des Registers I übernimmt die geänderte Initialisierungsroutine. Sie schreibt auch die Startadresse der Funktionstastenroutine in die Variable MIADD. Sie ist bei 65023 beziehungsweise 32255. Außerdem berücksichtigen die Änderungen, daß der Beginn des Basicprogramms sich verschieben kann.

Wie benutzt man die Routine? Genau wie bisher. Man schreibt sich REM-Zeilen, die die Befehle enthalten. Der Maschinencode kann nicht verschoben werden, da absolute Adressen verwendet werden. Man startet ihn mit »RANDOMIZE USR 65040« (32270) und man schaltet ihn ab mit »RANDOMIZE US 65030« (32260). Um weiteres braucht man sich nicht zu kümmern. Der Code kann mit »SAVE "name" CODE 65030,160« (32260,160) gesichert werden. Außerdem darf man vor dem Laden das »CLEAR 65022« (32254) nicht vergessen. Damit ist dann auch MIADD geschützt.

Microdrive-Hinweise

Noch ein Tip für Microdrive-Besitzer. Bei mir hat sich die folgende Belegung bewährt.

```
1 REM CAT: CAT 1@
2 REM LOAD: LOAD *"m";1;"
3 REM SAVE:INPUT a$:ERASE "m";1;a$:
SAVE *"m";1;a$@
```

In Zeile zwei habe ich das Zeichen »@« nicht vergessen. Ich brauche damit nur noch den Namen des Files einzutippen. Das »@« bewirkt, daß eine Folge von Befehlen ausgeführt wird; ohne das »@« kann man noch etwas anhängen. Und man muß dann ENTER drücken.

(R. W. Gerling)

```
65030   3E3EED47ED56C905  -> 961
65038   5D003EFDED4721FF  -> 1004
65046   FD362A2336FEED5E  -> 1023
65054   2A535C220DFE3E00  -> 580
65062   320FFEC9FFF3DDE5  -> 1468
65070   E5D5F53A0FFEFE01  -> 1269
65078   283BFDCB016E2861  -> 803
65086   CD1EFEDD2A0DFEDD  -> 1240
65094   7E04FEEA2050DD7E  -> 1077
65102   0521085CBE2814DD  -> 609
65110   6E02DD6603ED5B0D  -> 779
65118   FE191104001922OD  -> 372
65126   FE18D8CD89FE3E01  -> 1153
65134   320FFE182CDD2A0D  -> 663
65142   FEDD7E07FE0D281E  -> 945
65150   FE4020023E0DCD8C  -> 772
65158   FE1816DD7E073208  -> 712
65166   5C213B5CCBEE2A0D  -> 772
65174   FE23220DFEC9CD1E  -> 1026
65182   FEF1D1E1DDE1FBC9  -> 1827
```

Hex-Listing für 48-KByte-Version

```
               L5         :
FE41 DD2A0DFE  LD      IX,(L10     )
FE45 DD7E04    LD      A,(IX+4)
FE48 FEEA      CP      EA
FE4A 2050      JR      NZ,+80■L49
FE4C DD7E05    LD      A,(IX+5)
FE4F 21085C    LD      HL,LAST-K
FE52 BE        CP      (HL)
FE53 2814      JR      Z,+20■L25
FE55 DD6E02    LD      L,(IX+2)
FE58 DD6603    LD      H,(IX+3)
FE5E ED5B0DFE  LD      DE,(L10     )
FE5F 19        ADD     HL,DE
FE60 110400    LD      DE,0004
FE63 19        ADD     HL,DE
FE64 220DFE    LD      (L10     ),HL
FE67 18D8      JR      -40■L5
               L25        :
FE69 CD89FE    CALL    L45
FE6C 3E01      LD      A,01
FE6E 320FFE    LD      (L20     ),A
FE71 182C      JR      +44■L50
               L30        :
FE73 DD2A0DFE  LD      IX,(L10     )
FE77 DD7E07    LD      A,(IX+7)
FE7A FE0D      CP      0D
FE7C 281E      JR      Z,+30■L49
FE7E FE40      CP      40
FE80 2002      JR      NZ,+2■L31
FE82 3E0D      LD      A,0D
               L31        :
FE84 CD8CFE    CALL    L46
FE87 1816      JR      +22■L50
               L45        :
FE89 DD7E07    LD      A,(IX+7)
               L46        :
FE8C 32085C    LD      (LAST-K),A
FE8F 213B5C    LD      HL,FLAGS
FE92 CBEE      SET     5,(HL)
FE94 2A0DFE    LD      HL,(L10     )
FE97 23        INC     HL
FE98 220DFE    LD      (L10     ),HL
FE9B C9        RET
               L49        :
FE9C CD1EFE    CALL    L1
               L50        :
FE9F F1        POP     AF
FEA0 D1        POP     DE
FEA1 E1        POP     HL
FEA2 DDE1      POP     IX
FEA4 FB        EI
FEA5 C9        RET

Symbole:

LAST-K  EQU  5C08  (23560)
FLAGS   EQU  5C3B  (23611)
PROG    EQU  5C53  (23635)
MIADD   EQU  FDFF  (65023)
ABSCH   EQU  FE06  (65030)
L10     EQU  FE0D  (65037)
L20     EQU  FE0F  (65039)
ANSCH   EQU  FE10  (65040)
L1      EQU  FE1E  (65054)
START   EQU  FE2A  (65066)
L5      EQU  FE41  (65089)
L25     EQU  FE69  (65129)
L30     EQU  FE73  (65139)
L31     EQU  FE84  (65156)
L45     EQU  FE89  (65161)
L46     EQU  FE8C  (65164)
L49     EQU  FE9C  (65180)
L50     EQU  FE9F  (65183)
```

Assembler-Listing »Funktionstasten«

Spectrum-Hexerei

Hexadezimal-Listings sind kürzer und leichter einzutippen, als in dezimaler Form.

Um Ihnen das Abtippen zu erleichtern, haben wir ein spezielles Programm für die Ein- und Ausgabe mit Prüfsumme entwickelt.

Die Werte für die Zeilen 20 und 410 sind hier nur ein Beispiel. Diese Zahlen entnehmen Sie bitte dem jeweiligen Listing. Die Zahl rechts ist eine Prüfsumme, die automatisch errechnet wird und der Kontrolle dient.

Die Eingabe erfolgt mit RUN, die Ausgabe mit »RUN 400«. Spectrum-Listings, die Sie uns anbieten, müssen mit diesem Programm gelistet sein. (mk)

```
  0>REM RESERVIERTER PLATZ FUER
56 BYTE MASCHINENCODE-PROGRAMME

  1 REM
  2 REM    H E X  L O A D E R
  3 REM
  4 REM    SPECTRUM - VERSION
  5 REM    MANFRED-D. KOTTING
  6 REM
 10 REM SPEICHEREINGABE
 20 FOR I=23760 TO 23808 STEP 8
 30 PRINT I;"  ";
 40 INPUT A$
 41 IF LEN A$<>16 THEN GO TO 20
0
 45 PRINT A$;" -> ";
 46 LET C=0
 50 FOR J=I TO I+7
 60 LET P=(J-I)*2+1
 70 LET Y=CODE A$(P)
```

```
 71 IF Y>57 THEN LET Y=Y-7
 72 LET X=CODE A$(P+1)
 73 IF X>57 THEN LET X=X-7
 74 LET X=X-48+(Y-48)*16
 80 LET C=C+X
 90 POKE J,X
100 NEXT J
110 INPUT Z
115 IF C=Z THEN PRINT C
120 IF Z<>C THEN GO TO 200
130 NEXT I
140 STOP
200 CLS : REM FEHLERMELDUNG
210 PRINT "ERROR - GEBEN SIE DI
E ZEILE NOCHEINMAL EIN"
220 GO TO 30
400 REM SPEICHERAUSGABE
410 FOR I=23760 TO 23808 STEP 8
420 LET C=0
430 PRINT I;"  ";
440 FOR J=I TO I+7
450 LET A$=CHR$ INT (PEEK J/16+
48): LET B$=CHR$ (PEEK J-16*INT
(PEEK J/16)+48)
451 IF CODE A$>57 THEN LET A$=C
HR$ INT (PEEK J/16+55)
452 IF CODE B$>57 THEN LET B$=C
HR$ (PEEK J-16*INT (PEEK J/16)+5
5)
455 PRINT A$;B$;
460 LET C=C+PEEK J
470 NEXT J
475 PRINT " -> ";C
480 NEXT I
490 STOP
500 SAVE "HEXLOADER"

23760   5245534552564945  -> 613
23768   5254445522050 4C41  -> 570
23776   545A204655455220  -> 544
23784   3536204259544520  -> 479
23792   4D41534348494E45  -> 584
23800   4E434F44452D5052  -> 568
23808   4F4752414D4D4520  -> 552
```

Listing mit Testausdruck

Das Kassetten-Interface

Der Spectrum hat ein ausgezeichnetes Kassetten-Interface. Es ist allerdings nur sehr wenig über die Benutzung der entsprechenden ROM-Routinen bekannt.

Ein Kassetten-File besteht aus zwei Teilen: dem Header und dem Datenteil. Der Header ist 17 Byte lang. Dazu kommt noch am Anfang ein Flagbyte und am Schluß ein Checksummenbyte. Was in den 17 Byte steht, zeigt die folgende Aufstellung:

Byte	Inhalt
1	Typ: 0 = Programm 1 = Zahlen Array 2 = String Array 3 = Code
2—11	Die 10 Zeichen des Namens
12/13	Die Länge
14/15	Hängt ab vom Typ 0: Die Autostart Zeilennummer; Falls keine, den Wert 32768=8000h 1: Das erste Byte des Arrays: 2: Das erste Byte des Arrays: 3: Startadresse des Codes
16/17	Nur bei Programmen: Die Länge vom Programm ohne Variablen

Listing 1 zeigt ein Programm, das diese Information auswertet. Listing 2 zeigt einen typischen Ausdruck dieses Programms. Lediglich die Information aus Byte 16/17 wird nicht angezeigt. Sie wird aber im Programm in Zeile 200 berechnet (Variable s). Man kann sie also leicht einbauen, wenn man möchte.

Schauen wir uns in Listing 3 das Maschinencode-Programm an. Es ist im wesentlichen aus dem ROM kopiert. Es ruft die Routine an Adresse 1366=556h auf. Das ist eine Routine, die das Flagbyte testet, dann Bytes lädt und dann die Checksumme überprüft. Falls irgendetwas nicht stimmt, springt es mit Carryflag Reset zurück. Das kleine Maschinencode-Programm läuft in einer Endlos-Schleife, bis das Carryflag gesetzt ist. Das Flagbyte ist:

255=FFh bei Datenblock

0=00h bei Headerblock.

Das Flagbyte muß, bevor man die Routine 556h aufruft, in Register A stehen. Das Registerpaar DE enthält die Länge des zu lesenden Blocks (17 für einen Header). Im Registerpaar IX muß die Startadresse der Daten stehen.

Es gibt eine entsprechende Routine an der Adresse 1218=4C2h, die die Bytes saved. Die Parameter werden genauso übergeben, wie beim Laden.

Man kann ganz einfach Header-Lose Files erzeugen. Listing 4, 5 und 6 zeigen die SAVE-, LOAD- und VERIFY-Routinen. Die Parameter für START und LÄNGE sind so gesetzt, daß man mit den Programmen den Bildschirminhalt speichern und laden kann. Man sollte die Routinen aus kleinen Programmen benutzen. Interaktiv kann es Probleme beim VERIFY geben. Will man Header erzeugen, so muß das Register A mit 00h anstelle von FFh geladden werden.

Die Listings 4, 5 und 6 können an jede beliebige Stelle des RAMs gesetzt werden. Listing 7 zeigt alle drei Routinen im Happy-Computer-Format ab Adresse 50000. Die Einsprungadressen sind:

50000 LOAD
50016 SAVE
50047 VERIFY

Viel Spaß mit den Header-losen Files, die es sonst ja nur in gekauften Profi-Programmen gibt.

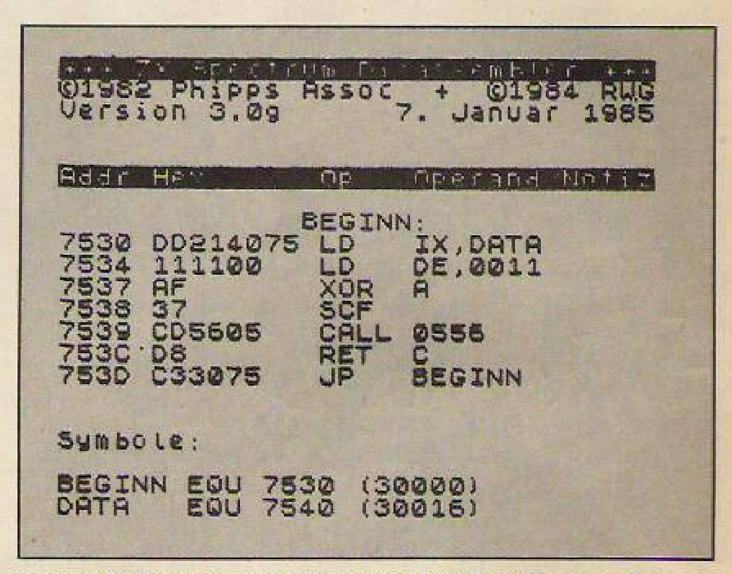
(R. W. Gerling)

```
  10 REM Tape Analyse
         © 1985 by
         Rainer W. Gerling
  11 CLEAR 29999
  15 DEF FN p(x)=PEEK x+256*PEEK
 (x+1): LET p=30016
  20 FOR i=1 TO 16: READ a: POKE
 29999+i,a: NEXT i
  30 DATA 221,33,64,117,17,17,0,
175,55,205,86,5,216,195,48,117
  40 PRINT INVERSE 1;"* Tape Ana
lyse by R.W. Gerling *"; INVERSE
 0
  50 PRINT "  Name";TAB 10;"Laen
ge   Bemerkung"'': PLOT 0,156: D
RAW 255,0
  60 PRINT #1;" Start Tape, then
 press any key": PAUSE 0
  65 PRINT #1;AT 1,0,,
  70 RANDOMIZE USR 30000
  75 LET n$=""
  80 LET typ=PEEK p: LET l=FN p(
p+11): FOR i=p+1 TO p+10: LET n$
=n$+CHR$ PEEK i: NEXT i
  90 LET l$=STR$ l: LET l$="
 "(1 TO 6-LEN l$)+l$
 100 PRINT n$;l$;" ";
 110 IF typ>3 THEN PRINT "Lesefe
hler": GO TO 70
 130 IF typ<3 THEN GO TO 170
 140 LET s=FN p(p+13)
 150 IF s=16384 AND l=6912 THEN
PRINT "SCREEN$ ": GO TO 70
 160 PRINT "CODE ab: ";s: GO TO
70
 170 IF typ=0 THEN GO TO 200
 180 LET s=PEEK (p+14): LET t=IN
T (s/32): LET l$=CHR$ (s-32*t+96
): IF t=2 OR t=6 THEN LET l$=l$+
"$"
 190 PRINT "Array: ";l$: GO TO 7
0
 200 LET l=FN p(p+13): LET s=FN
p(p+15)
 210 PRINT "Prog.";
 220 IF l<10000 THEN PRINT " LIN
E ";l;
 230 PRINT : GO TO 70
9999 SAVE *"m";1;"T-Analyse": VE
RIFY *"m";1;"T-Analyse"
```

Listing 1. Das Programm Tape-Analyse listet alle Files auf einem Band.

```
* Tape Analyse by R.W. Gerling *
  Name      Laenge  Bemerkung
------------------------------------
Delete       1132 Prog.
Extime        356 Prog.
Memomap      1072 Prog.
mctt 16      2354 Prog. LINE 4
zxdisasm     7948 Prog.
zxmcmon      2635 Prog.
toolkit 1     256 CODE ab: 32500
zxdisasm     8471 Prog.
zxasm        8241 Prog.
errortest     774 Prog.
errorprog    2026 Prog.
Sound-Demo   2213 Prog.
tasword      6728 Prog. LINE 15
tasword     10751 CODE ab: 54784
```

Listing 2. Beispiel Ausdruck des Programms Tape-Analyse.



```
+++ ZX Spectrum Disassembler +++
©1982 Phipps Assoc   +   ©1984 RWG
Version 3.0g        7. Januar 1985

Addr Hex        Op    Operand Notiz

                BEGINN:
7530 DD214075   LD    IX,DATA
7534 111100     LD    DE,0011
7537 AF         XOR   A
7538 37         SCF
7539 CD5605     CALL  0556
753C D8         RET   C
753D C33075     JP    BEGINN

Symbole:

BEGINN EQU 7530 (30000)
DATA   EQU 7540 (30016)
```

Listing 3. Disassemblerlisting der MC-Routine aus dem Programm Tape-Analyse.

```
+++ ZX Spectrum Disassembler +++
©1982 Phipps Assoc   +   ©1984 RWG
Version 3.0g        7. Januar 1985

Addr Hex        Op    Operand Notiz

                SAVEN   :
C350 3EFD       LD    A,FD
C352 CD0116     CALL  1601
C355 AF         XOR   A
C356 11A109     LD    DE,09A1
C359 CD0A0C     CALL  0C0A
C35C DDCB02EE   SET   5,(IX+2)
C370 CDD415     CALL  15D4
C373 11001B     LD    DE,LAENGE
C376 D9         EXX
C377 210040     LD    HL,START
C37A 3EFF       LD    A,FF
C37C C3C204     JP    04C2

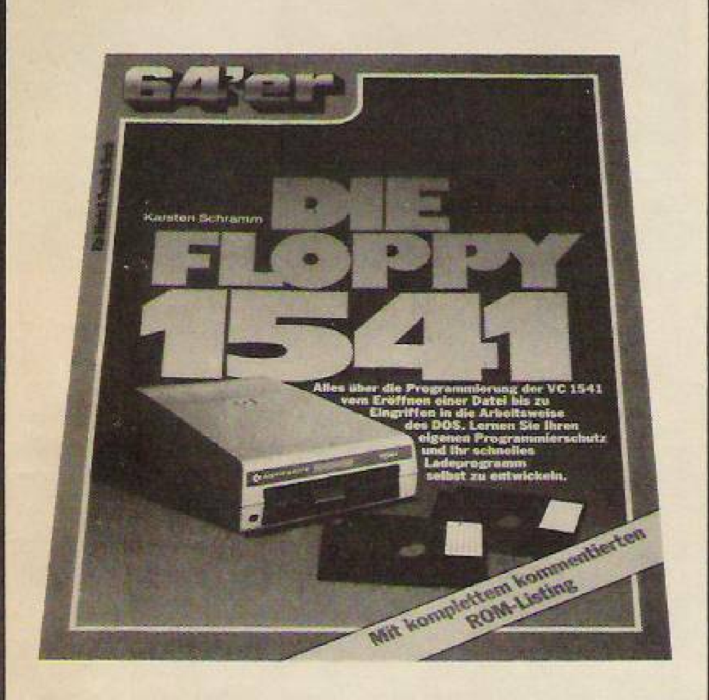
Symbole:

LAENGE EQU 1B00 ( 6912)
START  EQU 4000 (16384)
SAVEN  EQU C350 (50016)
```

Listing 4. Ein Maschinencode-Programm, das den Bildschirminhalt ohne Header auf Band speichert. Die Werte für LÄNGE und START müssen für andere Anwendungen geändert werden. Die Routine bei 3082=C0Ah druckt die Start Tape Message und die Routine bei 5588=15D4h wartet, bis eine Taste gedrückt wird.

Markt&Technik-Buchverlag

Werden Sie ein Profi — mit dem Commodore 64



K. Schramm

Die Floppy 1541 April 1985, 434 Seiten
Für alle Programmierer, die mehr über ihre VC 1541-Floppystation erfahren wollen. Der Vorgang des Formatierens · das Schreiben von Files auf Diskette · die Funktionsweise von schnellen Kopier- und Ladeprogrammen · viele fertige Programme · Lesen und Beschreiben von defekten Disketten · Für Einsteiger und für fortgeschrittene Maschinensprache-Programmierer.
Best.-Nr. MT 806, ISBN 3-89090-098-4 — **DM 49,— (Sfr. 45,10/öS 382,20)**
Best.-Nr. MT 710 (Beispiele auf Diskette) — **DM 29,90* (Sfr. 29,90/öS 269,10)**
*** inkl. MwSt. Unverbindliche Preisempfehlung**

Der sensible Commodore 64. Januar 1985, 144 Seiten
Eine Softwaresammlung zu den technologischen Neuerscheinungen im Commodore 64 · für Erstbenutzer wie für Experten · ein Buch der Softwarenutzung aller technologischen Eigenheiten des C-64.
Best.-Nr. PW 727 — **DM 29,80 (Sfr. 27,50/öS 232,40)**

Commodore 64 Programmsammlung. Januar 1985, 200 Seiten
Vielfältig wie die Interessen am Commodore 64: eine kommentierte Programmsammlung für Spiel, Computerwissen, Datenverwaltung und Kinder.
Best.-Nr. PW 726 — **DM 29,80 (Sfr. 27,50/öS 232,40)**

Mehr als 32 BASIC-Programme für den Commodore 64. Februar 1984, 279 Seiten
Programme speziell für den Commodore 64 · umfassende praktische Anwendungen · jede Menge Lehr- und Lernhilfen · super Spiele · für Basic-Neulinge und Experten.
Best.-Nr. MT 613 (Buch) — **DM 49,— (Sfr. 45,10/öS 382,20)**
Best.-Nr. MT 614 (Beispiele auf Diskette) — **DM 48,— * (Sfr. 48,—/öS 432,—)**
*** inkl. MwSt. Unverbindliche Preisempfehlung**

MSX Basic. April 1985, 236 Seiten
Alles über den neuen Heimcomputerstandard MSX: zusätzlich zum »normalen« BASIC können mit insgesamt mehr als 150 Befehlen und Funktionen Grafiken erstellt, Töne erzeugt, Melodien komponiert und ganze Spielhandlungen programmiert werden · 32 Sprites garantieren abwechslungsreiche Action-Spiele · die Hardware des MSX-Systems · nützliche Hinweise zur Dateibehandlung · das MSX-BASIC anhand der Entwicklung eines Spielszenarios mühelos lernen · drei vollständige Spiele: Der eisige Planet, Autorennen und Bilder entwerfen · mit ausführlicher Befehlsübersicht · für Anfänger!
Best.-Nr. MT 805, ISBN 3-89090-107-7 — **DM 44,— (Sfr. 40,50/öS 343,20)**
Best.-Nr. 825 (Beispiele auf Kassette) — **DM 19,90* (Sfr. 19,90/öS 179,10)**
*** inkl. MwSt. Unverbindliche Preisempfehlung**

Computerchinesisch für Einsteiger. Juli 1984, 107 Seiten
Ein praxisnahes Lexikon, das Personal Computer-Benutzern und solchen, die es werden wollen, das Lesen von Fachzeitschriften, Büchern, Bedienungsanleitungen und Datenblättern erleichtert · über 1000 häufig benötigte Fachbegriffe klar und verständlich erläutert · mit zahlreichen Abbildungen.
Best.-Nr. MT 690 — **DM 28,— (Sfr. 25,90/öS 218,40)**

Die angegebenen Preise sind Ladenpreise.



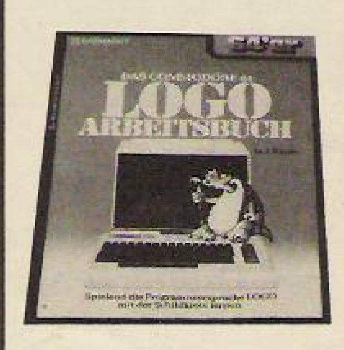
H. L. Schneider

Commodore 64 Listings Band 2: Dateiverwaltung · Schule Hobby
Oktober 1984, 179 Seiten
Ein Buch mit Programmen für die ganze Familie · DATAVE — Eine Dateiverwaltung · mathematische Funktionen · Konjugation und Deklination in Latein · Regressionsanalyse · Bundesligatabelle.
Best.-Nr. MT 766
(Sfr. 23,—/öS 193,40) **DM 24,80**



S. Urute

Grafik & Musik auf dem Commodore 64
Oktober 1984, 336 Seiten
Eine hervorragende Einführung in die Programmierung des Commodore 64, speziell des VIC.II-Grafikbaus sowie des eingebauten Synthesizers · 68 gut strukturierte und kommentierte Beispielprogramme zur Erzeugung von Sprites und Klangeffekten · Sprite-Tricks · Zeichengrafik · hochauflösende Grafik.
Best.-Nr. MT 743
(Sfr. 35,—/öS 296,40) **DM 38,—**



M. J. Winkler

Das Commodore 64-LOGO-Arbeitsbuch
September 1984, 225 Seiten
Kinder lernen auf dem Commodore 64 mit der Schildkröte als Lehrer: Bilder malen · Grafikeffekte erzeugen · Wörter verarbeiten · Prozeduren und Variablen · Umgang mit Begriffen wie: Längenmaß, Winkel, Dreieck, Quadrat.
Best.-Nr. MT 720
(Sfr. 31,30/öS 265,20) **DM 34,—**



Dr. P. Albrecht

Commodore 64 — Multiplan
März 1984, 230 Seiten
Multiplan jetzt auch für den Commodore 64 · der volle Leistungsumfang der 16-Bit-Version · Einführung in die Arbeitsweise von Tabellenkalkulationsprogrammen · praxisnahe Beispiele · Beschreibung alle Befehle und Funktionen · nicht nur für Anfänger.
Best.-Nr. MT 655
(Sfr. 44,20/öS 374,40) **DM 48,—**

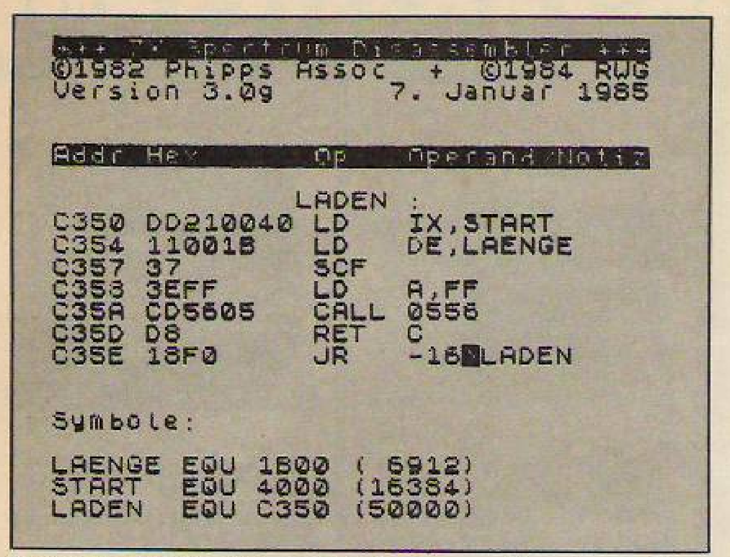
Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Fragen Sie dort nach unserem Gesamtkatalog mit über 170 neuen Computerbüchern.



Bestellkarten bitte an Ihren Buchhändler oder an einen unserer Depot-Händler. Adressenverzeichnis am Ende des Heftes. Beim Markt & Technik Verlag eingehende Bestellungen werden von den Depot-Händlern ausgeliefert.

Markt & Technik Verlag AG Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik-Vertriebs AG, Kollerstraße 3, CH-6300 Zug, ☎ 042/22 31 55
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien ☎ 0222/67 75 26



```
+++ ZX Spectrum Disassembler +++
©1982 Phipps Assoc  +  ©1984 RWG
Version 3.0g       7. Januar 1985

Addr Hex      Op   Operand Notiz

              LADEN :
C350 DD210040 LD   IX,START
C354 11001B   LD   DE,LAENGE
C357 37       SCF
C358 3EFF     LD   A,FF
C35A CD5605   CALL 0556
C35D D8       RET  C
C35E 18F0     JR   -16 LADEN

Symbole:

LAENGE EQU 1B00 ( 6912)
START  EQU 4000 (16384)
LADEN  EQU C350 (50000)
```

Listing 5. Ein Maschinencode-Programm, das einen mit Listing 4 geSAVEten Bildschirminhalt wieder lädt.

```
+++ ZX Spectrum Disassembler +++
©1982 Phipps Assoc  +  ©1984 RWG
Version 3.0g       7. Januar 1985

Addr Hex      Op   Operand Notiz

              VERIFY:
C37F DD210040 LD   IX,START
C383 11001B   LD   DE,LAENGE
C386 A7       AND  A
C387 3EFF     LD   A,FF
C389 CD5605   CALL 0556
C38C D8       RET  C
C38D CF       RST  08
C38E 1A       DEFB 1A

Symbole:

LAENGE EQU 1B00 ( 6912)
START  EQU 4000 (16384)
VERIFY EQU C37F (50047)
```

Listing 6. Ein Maschinencode-Programm, das einen mit Listing 4 geSAVEten Bildschirminhalt VERIFYed.

```
50000  DD21004011001B37 -> 417
50008  3EFFCD5605D818F0 -> 1093
50016  3EFDCD0116AF11A1 -> 896
50024  09CD0A0CDDCB02EE -> 900
50032  CDD415110018DD21 -> 736
50040  00403EFFC3C204DD -> 995
50048  21004011001BA73E -> 370
50056  FFCD5605D8CF1A00 -> 1000
```

◀ Listing 7. Das Hex-Listing aller Routinen

Quelle für ROM Routinen:
Ian Logan und Frank O'Hara, The Complete Spectrum ROM Disassembly, Melbourne House Publishers, Tring 1983

Mathe-Trainer

Es handelt sich hier um ein Übungsprogramm für die vier Grundrechenarten.

Das gesamte Programm ist recht gegliedert geschrieben, so daß kaum Schwierigkeiten auftreten, den Programmablauf zu verfolgen. Doch sei folgend noch auf die Funktion einzelner Unterprogramme eingegangen.

Umlaute

Gleich nach Programmstart wird dieser Programmteil ausgeführt. Da die Buchstaben a, o, u und s im Alphabet nicht aufeinander folgen, die Umlaute aber in den dazugehörigen UDGs untergebracht werden sollten, wurde der Umweg über den String gewählt.

Hauptprogramm

Das Hauptprogramm stellt auf dem Bildschirm ein Menü dar, läßt eine Melodie erklingen und verzweigt den Programmlauf je nach Wahl in die einzelnen Funktionen.

GET i$

Etwas Besonderes stellt dieses Unterprogramm dar. Eigentlicher Sinn ist es, wie der Name schon sagt, die Simulation des GET-Befehls einiger Basic-Interpreter. Doch geschieht noch einiges mehr:

— Der Text in t$ wird in Bildschirmzeile 23 als Laufschrift dargestellt.

— Der Cursor (heller Untergrund) wird an der Bildstelle (y,x) dargestellt.

— Die Eingabe wird überprüft, ob sie in M$ (für Menü) enthalten ist.

Weiterhin sei noch erwähnt, daß in diesem Unterprogramm nicht INKEY$, sondern die Systemvariable LAST-K aus Reaktionsgründen verwendet wurde.

Rahmen

Zur Verzierung sind an einigen Programmstellen Sternchenumrandungen verwendet worden. Hier werden sie erzeugt.

CLS

Die Tastatur wird auf CAPS-LOCK eingestellt, der Bildschirm gelöscht und die Kopfzeile ausgegeben.

Falsch

Bei falscher Eingabe innerhalb der Rechenaufgaben wird dieses Unterprogramm angesteuert. Es läßt eine Melodie erklingen und ändert den Laufschrifttext. Weiterhin wird die Fehleranzahl (fe) inkrementiert.

Richtig

Ist eine Ziffer richtig, dann wird diese Prozedur ausgeführt. Eine kurze Tonfolge erklingt, die mögliche Gesamtfehleranzahl wird inkrementiert und der Laufschrifttext geändert.

Musik

Diese Prozedur liest die DATA-Zeile, deren Nummer in »mel« gespeichert ist und spielt sie als Melodie. Da der Spectrum keine Musikpausen kennt, wurde der höchstmögliche Ton verwendet, den wohl kaum ein menschliches Ohr hören kann, zumal die Lautstärke des Spectrum nicht gerade das Trommelfell platzen läßt. Weiterhin wird während der Musik die Tastatur abgefragt und die Prozedur endet, sobald eine Taste gedrückt wird. Um jedoch weder durch vorige Tastendrucke diesen Vorgang auszulösen noch diesen Tastendruck an die weitere Tastaturabfragen zu übergeben, sind noch einige Sicherheitsvorkehrungen getroffen worden.

Addition

Dieses Unterprogramm stellt eine Additions-Aufgabe und erwartet die einzelnen Ziffern der schriftlichen Lösung, worauf dann die Korrektheit oder Unkorrektheit angezeigt und durch eine Tonfolge bekräftigt wird. Nach der Ausgabe der Überschrift, einem erläuternden Text und einem Rahmen für die Aufgabe, ertönt eine Empfangsmelodie. Dann beginnt eine Laufschrift, und das Rechnen kann starten. Die Annahme einer Ziffer gliedert sich in drei Teile:

— Berechnen des Wertes und Umformen in einen String

— Lesen einer Ziffer von der Tastatur

— Korrektheit prüfen und eventuell neu lesen

Die Berechnung des Übertrages ist der Summenberechnung gleich angeschlossen, also ist der Ablauf berechnen, be-

```
*********** M e n ü ************
*                              *
*   A)ddition                  *
*   S)ubtraktion               *
*   M)ultiplikation            *
*   D)ivision                  *
*   T)est                      *
*   B)eendigung des Programms  *
*                              *
* Bitte  den Anfangsbuchstaben *
* der  gewünschten   Funktion  *
* eingeben!                    *
*                              *
********************************
```

Das Hauptmenü

```
Angefangen mit den *************
Einern,  also  der *           *
letzten    Spalte, *           *
sollen die Ziffern *    00472  *
addiert werden.    *  + 12160  *
Die   Einer    der *  + 69557  *
Summe  werden dann *  + 56960  *
unter den  Doppel= *  -------  *
strich der betref= *           *
fenden Spalte, die *  =======  *
Zehner  über   den *           *
Doppelstrich   der *           *
nächsten    Spalte *           *
links geschrieben. *************
Bei   den  Zehnern
bis  Zehntausendern müssen  dann
auch  die  Zehner  der   letzten
Spalte mit addiert werden.
```

Additionsübung

rechnen, lesen, prüfen, lesen, prüfen. Überträge werden nur dann von der Tastatur erwartet, wenn sie nicht Null sind.

Subtraktion, Multiplikation, Division

Der Ablauf und die Methode sind mit denen der Addition zu vergleichen.

Test

Dieses Unterprogramm liest den Namen des zu testenden Schülers, stellt von jedem Aufgabentyp eine Aufgabe, und gibt eine Urkunde aus, die auch ausgedruckt werden kann.

Anleitung zur Benutzung

Bevor ein Schüler an den Computer gesetzt wird, um mit diesem Programm das schriftliche Berechnen der vier Grundrechnungsarten zu erlernen, sollte dies die ältere Schwester, der alte Bruder, ein Elternteil oder eine andere Person tun, die das Rechnen beherrscht und sich die erläuternden Texte neben den Aufgaben durchlesen. Denn die Reihenfolge des Niederschreibens der einzelnen Ziffer kann von Lehrer zu Lehrer schon recht verschieden sein, und diese Reihenfolge ist auch einzuhalten, da das Programm die Eingabe sonst als falsch wertet. Doch dürften dahingehend eigentlich kaum Schwierigkeiten entstehen, weil das Programm an die Stelle, an der eine Eingabe erwartet wird, einen Cursor setzt.

Variable	Bedeutung
xa, ya	Koordination des oberen linken Bildpunktes für Rahmen
dx,dy	Ausdehnung des Rahmens in x- und y-Richtung
t	Nummer des ersten Zeichens auf dem Bildschirm im Laufschrifttext
t$	Laufschrifttext
mel	Zeile, in der die zu spielende Medolie in DATA-Statements untergebracht ist
ges	zählt die möglichen Fehler
fe	zählt die gemachten Fehler
m$	gibt die möglichen eingaben für GET an
i$	übergibt das in GET gelesene Zeichen
y,x	übergibt die Position, an der das Zeichen ausgegeben werden soll
z$	speichert die Ziffern einer Aufgabe und deren Zwischenergebnisse
s$	Summe einer Spalte
u$	Übertrag
d$	Differenz einer Spalte
q$	Ergebnis der Ganzzahldivision
p$	Produkt zweier Ziffern
r$	Rest der Division

Variablenliste

```
  10 REM
  11 REM
  12 REM [illegible]
  13 REM
  20 RESTORE
  30 LET u$="aous"
  40 FOR n=1 TO 4: FOR m=0 TO 7
  41 READ a: POKE USR u$(n)+m,a
  42 NEXT m: NEXT n
  50 DATA 40,0,56,4,60,68,60,0
  51 DATA 40,0,56,68,68,68,56,0
  52 DATA 40,0,68,68,68,68,56,0
  53 DATA 0,112,72,112,72,68,120
64
 100 REM
 101 REM
 102 REM [illegible]
 103 REM
 110 GO SUB 400
 120 LET xa=3: LET dx=26: LET ya
=2: LET dy=5: GO SUB 300
 121 PRINT AT 2,9;" Version  1,5
";AT 4,9;"Michael Hönnig";AT 6,
12;" © 1984 "
 130 LET xa=0: LET dx=32: LET ya
=8: LET dy=14: GO SUB 300
 131 PRINT AT 8,11;" M e n ü ";A
T 10,3;"A)ddition";AT 11,3;"S)ub
traktion";AT 12,3;"M)ultiplikati
on";AT 13,3;"D)ivision";AT 14,3;
"T)est";AT 15,3;"B)eendigung des
 Programms";AT 17,2;"Bitte  den
Anfangsbuchstaben ** der  gewün
schten  Funktion ** eingeben!"
 132 LET t=1: LET t$="          Di
e einzelnen Rechenmethoden sollt
en dem Schüler zuvor erklärt wer
den. Zu diesem Zweck sind neben
den Aufgaben kurze Anleitungen g
egeben.    Viel Spaß und ebensov
iel Erfolg wünscht Michael.     "
 133 LET mel=810: GO SUB 900
 134 LET ges=0: LET fe=0
 140 LET m$="ASMDTB": LET y=19:
LET x=12: GO SUB 200
 141 IF i$="A" THEN GO SUB 1000
 142 IF i$="S" THEN GO SUB 2000
 143 IF i$="M" THEN GO SUB 3000
 144 IF i$="D" THEN GO SUB 4000
 145 IF i$="T" THEN GO SUB 5000
 146 IF i$<>"B" THEN RUN
 150 GO SUB 400
 151 LET xa=7: LET dx=18: LET ya
=9: LET dy=5: GO SUB 300: PRINT
AT 11,13;"tschüß"
 160 POKE 23658,0
 199 STOP
 200 REM
 201 REM
 202 REM [illegible]
 203 REM
 210 PRINT AT y,x; BRIGHT 1;" "
 220 IF INKEY$<>"" THEN GO TO 22
0
 221 POKE 23560,0
 230 LET i$=CHR$ PEEK 23560
 231 LET t=t+.5: IF t>LEN t$ THE
N LET t=1
 232 IF INT t+31<=LEN t$ THEN PR
INT #0;AT 1,0;t$(INT t TO INT t+
31): GO TO 234
 233 PRINT #0;AT 1,0;t$(INT t TO
);t$( TO 31-LEN t$+INT t)
 234 IF i$=CHR$ 0 THEN GO TO 230
 240 FOR i=1 TO LEN m$
 241 IF m$(i)=i$ THEN GO TO 250
 242 NEXT i: IF i$="B" THEN RUN
100
 243 BEEP .5,-10: GO TO 220
 250 PRINT AT y,x;i$
 251 BEEP .01,-20
 260 RETURN
 300 REM
 301 REM
 302 REM [illegible]
```

Basic-Listing »Mathe«

```
 303 REM
 310 PRINT AT ya,xa;
 320 FOR x=1 TO dx: PRINT "*";:
NEXT x
 330 PRINT AT ya+dy-1,xa;
 340 FOR x=1 TO dx: PRINT "*";:
NEXT x
 350 FOR y=1 TO dy-2: PRINT AT y
a+y,xa;"*";TAB xa+dx-1;"*": NEXT
y
 360 RETURN
 400 REM
 401 REM
 402 REM *** CLS ***
 410 CLS
 420 POKE 23658,8
 430 PRINT "GRUNDRECHENARTEN UEB
UNGSPROGRAMM"
 440 RETURN
 500 REM
 501 REM
 502 REM *** falsch ***
 503 REM
 510 PRINT #0;AT 1,0; FLASH 1;"
      Leider falsch!",
 520 FOR t=0 TO -10 STEP -2.5
 521 BEEP .2,t: BEEP .2,t: BEEP
2,t
 522 NEXT t: BEEP .6,t
 530 LET t$="          Leider war
dies falsch. Aber rechne doch no
ch einmal nach - vielleicht erhä
ltst Du dann das richtige Ergebn
is.     ": LET t=1 .
 540 LET fe=fe+1
 550 RETURN
 600 REM
 601 REM
 602 REM *** richtig ***
 603 REM
 610 BEEP .1,-7: BEEP .1,0
 620 LET t$="          Das war ric
htig. Auf zur nächsten Ziffer. V
iel Erfolg!     ": LET t=1
 640 LET ges=ges+1
 650 RETURN
 700 REM
 701 REM
 702 REM *** Aufgabe richtig ***
 703 REM
 710 PRINT #0;AT 1,0; FLASH 1;"
 Alles richtig - gut gemacht!  "
 720 LET mel=990: GO SUB 900
 730 PAUSE 200
 740 RETURN
 800 REM
 801 REM
 802 REM [illegible]
 803 REM
 810 LET t=1: LET t$="          We
nn Dir klar ist, wie Du die Aufg
abe rechnen mußt, dann kannst Du
 anfangen, sonst frage jemanden,
 der es weiß, z. B. deine Eltern
.    Um diese Aufgabe zu beenden
, gebe einfach 'B' ein.      "
 811 RETURN
 900 REM
 901 REM
 902 REM *** Musik ***
 903 REM
 910 RESTORE mel
 911 PRINT #0;AT 1,0;" Jede Tast
e beendet die Musik!  "
 920 IF INKEY$<>"" THEN GO TO 92
0
 930 IF INKEY$="" THEN GO TO 940
 931 IF INKEY$<>"" THEN GO TO 93
1
 932 RETURN
 940 READ dur,pit
 941 IF dur<>0 THEN BEEP dur,pit
: GO TO 930
 942 IF pit<>0 THEN BEEP pit,69:
GO TO 930
 950 RETURN
 960 DATA .2,4,.2,7,.2,7,.2,7,.2
,9,.2,7,.2,7,.2,4,.2,7,.2,7,.2,9
,.2,7,.4,7,.2,5,0,.2,.2,7,.2,7,.
2,7,.2,7,.2,9,.2,7,.2,7,.2,2,.2,
7,.2,7,.2,9,.2,7,.4,7,.2,4,0,.2,
.2,7,.2,7,.2,7,.2,7,.2,9,.2,7,.2
,7,.2,7,.2,12,.2,12,.2,14,.2,12,
.4,12,.2,9,0,.2,.2,9,.2,12,.2,11
,.2,9,.2,12,.2,7,.2,7,.2,4,.2,7,
.2,7,.2,9,.2,11,.4,12,.2,12,0,0
 971 DATA .2,4,.2,2,.2,0,.2,2,.2
,4,.2,4,.4,4,.2,2,.2,2,.4,2,.2,4
,.2,7,.4,7,.2,4,.2,2,.2,0,.2,2,.
2,4,.2,4,.4,4,.2,2,.2,2,.2,4,.2,
2,.6,0,0,0
 972 DATA .2,4,.2,4,.2,5,.2,7,.2
,7,.2,5,.2,4,.2,2,.2,0,.2,0,.2,2
,.2,4,.2,4,.2,2,.4,2,.2,4,.2,4,.
2,5,.2,7,.2,7,.2,5,.2,4,.2,2,.2,
0,.2,0,.2,2,.2,4,.2,2,.2,0,.4,0,
0,0
 973 DATA .2,4,.2,7,.2,7,.2,7,.2
,5,.2,9,.4,9,.2,11,.2,11,.2,9,.2
,11,.2,12,.2,14,.4,16,.2,4,.2,7,
.2,7,.2,7,.2,5,.2,9,.4,9,.2,11,.
2,11,.2,9,.2,11,.2,12,.2,12,.2,1
2,0,0
 974 DATA .3,7,.1,9,.2,7,.2,5,.2
,4,.2,5,.4,7,.2,2,.2,4,.4,5,.2,4
,.2,5,.4,7,.3,7,.1,9,.2,7,.2,5,.
2,4,.2,5,.4,7,.2,2,.2,5,.2,4,.2,
2,.4,0,0,0
 980 DATA .3,7,.1,9,.1,7,.1,5,.2
,4,.2,0,.2,0,.2,2,.2,-5,.2,-5,.2
,4,.2,0,.2,0,.3,7,.1,9,.1,7,.1,5
,.2,4,.2,0,.2,0,.2,2,.2,-5,.2,-5
,.6,0,.2,2,.2,-5,.2,-5,.2,4,.2,0
,.2,0,.2,2,.2,-5,.2,-5,.2,4,.2,0
,.2,0,.3,7,.1,9,.1,7,.1,5,.2,4,.
2,0,.2,0,.2,2,.2,-5,.2,-5,.4,0,0
,0
 990 DATA .1,1,.1,-3,.2,-3,.1,1,
.1,-3,.2,-3,.1,-3,.2,-1,.1,-1,.1
,4,.1,4,.1,4,.2,1,.1,-3,.2,-3
 999 DATA 0,0
1000 REM
1001 REM
1002 REM *** Addition ***
1003 REM
1010 GO SUB 400: PRINT AT 2,8;"*
** Addition ***"
1011 LET xa=19: LET dx=13: LET y
a=4: LET dy=14: GO SUB 300
1012 PRINT AT 4,0;"Angefangen mi
t den"'"Einern,  also  der"'"let
zten    Spalte,"'"sollen die Zif
fern"'"addiert werden."'"Die
Einer   der"'"Summe   werden dann
"'"unter den  Doppel="'"strich d
er betref="'"fenden Spalte, die"
'"Zehner  über   den"'"Doppelstr
ich   der"'"nächsten     Spalte"'
"links geschrieben."'"Bei   den
 Zehnern"'"bis  Zehntausendern m
üssen  dannauch  die  Zehner  de
r   letztenSpalte mit addiert we
rden."
1020 DIM z$(5,6)
1021 FOR n=1 TO 4: FOR z=2 TO 6
1022 LET z$(n,z)=CHR$ INT (10*RN
D+48)
1023 NEXT z
1024 PRINT AT 6+n,22;"+" AND N>1
;" " AND N=1;z$(n): NEXT n
1025 LET z$(5)="000000"
1026 PRINT AT 11,22;"-------";AT
 13,22;"======="
1027 LET mel=971: GO SUB 900
1030 GO SUB 800
1100 FOR n=6 TO 2 STEP -1
1101 LET m$="0123456789"
1110 LET s=0: FOR m=1 TO 5: LET
s=s+CODE z$(m,n)-48: NEXT m
1111 LET s$=STR$ s: LET u$="0":
IF LEN s$=2 THEN LET u$=s$(1): L
ET s$=s$(2)
1120 LET y=14: LET x=22+n: GO SU
B 200
```

Basic-Listing »Mathe« (Fortsetzung)

```
1121 IF i$<>s$ THEN GO SUB 500:
GO TO 1120
1122 GO SUB 600
1123 IF U$="0" THEN GO TO 1133
1130 LET y=12: LET x=21+n: GO SU
B 200
1131 IF i$<>u$ THEN GO SUB 500:
GO TO 1130
1132 GO SUB 600
1133 LET z$(5,n-1)=u$
1140 NEXT n
1150 LET x=23: LET y=14: GO SUB
200
1151 IF i$<>z$(5,1) THEN GO SUB
500: GO TO 1150
1152 GO SUB 600
1170 GO SUB 700
1200 RETURN
2000 REM
2001 REM
2002 REM *** Subtraktion ***
2003 REM
2010 GO SUB 400: PRINT AT 2,6;"*
** Subtraktion ***"
2011 LET xa=19: LET dx=13: LET y
a=4: LET dy=14: GO SUB 300
2012 PRINT AT 4,0;"Die mit Minu
szei="'"chen    versehenen"'"Zah
len werden, mit"'"den Einern  b
egin"'"nend, addiert."'"Diese  S
umme  wird"'"von  der  obersten"
'"Ziffer der betref="'"fenden Sp
alte sub="'"trahiert;    dabei"'
"müssen  Zehner ge="'"borgt werd
en.  Die"'"zu  borgenden Zeh="'"
ner   werden   zu="'"nächst   Üb
er  der"'"Doppellinie  notiert,
dann  wird"'"die Differenz  unte
r die Doppel="'"linie geschriebe
n."
2020 DIM z$(5,6)
2021 FOR n=1 TO 4: FOR z=2 TO 6
2022 LET z$(n,z)=CHR$ INT (10*RN
D+48): NEXT z
2023 IF n=1 THEN LET z$(1,1)=CHR
$ INT (8*RND+50)
2024 PRINT AT 6+n,22;"-" AND n>1
;" " AND n=1;z$(n): NEXT n
2025 LET z$(5)="000000"
2026 PRINT AT 11,22;"-------";AT
 13,22;"======="
2027 LET mel=972: GO SUB 900
2030 GO SUB 800
2100 FOR n=6 TO 2 STEP -1
2101 LET m$="0123456789"
2110 LET s=0: FOR m=2 TO 5: LET
s=s+CODE z$(m,n)-48: NEXT m
2111 LET s$=STR$ s: IF LEN s$=1
THEN LET s$="0"+s$
2112 LET u$=s$(1): IF s$(2)>z$(1
,n) THEN LET u$=CHR$ (CODE s$(1)
+1)
2113 LET d$=STR$ (VAL (u$+z$(1,n
))-VAL s$)
2114 IF u$="0" THEN GO TO 2123
2120 LET y=12: LET x=21+n: GO SU
B 200
2121 IF i$<>u$ THEN GO SUB 500:
GO TO 2120
2122 GO SUB 600
2123 LET z$(5,n-1)=u$
2130 LET y=14: LET x=22+n: GO SU
B 200
2131 IF i$<>d$ THEN GO SUB 500:
GO TO 2130
2132 GO SUB 600
2140 NEXT n
2150 LET d$=STR$ (CODE z$(1,1)-C
ODE z$(5,1))
2151 LET y=14: LET x=23: GO SUB
200
2152 IF i$<>d$ THEN GO SUB 500:
GO TO 2151
2153 GO SUB 600
2160 GO SUB 700
2200 RETURN
3000 REM
3001 REM
3002 REM *** Multiplikation ***
3003 REM
3010 GO SUB 400: PRINT AT 2,5;"*
** Multiplikation ***"
3011 LET xa=19: LET dx=13: LET y
a=4: LET dy=14: GO SUB 300
3012 PRINT AT 4,0;"Zunächst    w
erden"'"die Hunderter  der"'"zwe
iten  Zahl  mit"'"den Einern der
 an="'"deren multiplizie="'"rt,
die  Einer des"'"Produkts   noti
ert"'"und die Zehner ge="'"merkt
.  Nun werden"'"die Hunderter  d
er"'"zweiten   mit  den"'"Zehner
n der ersten"'"multipliziert  un
d"'"zur  oben germerk="'"ten Zah
l addiert.."'"Die dabei entstehe
nden drei Zah=len werden  wie  b
ei  'Addition'addiert."
3020 DIM z$(5,6)
3021 FOR z=1 TO 6
3022 LET z$(1,z)=CHR$ INT (10*RN
D+48): NEXT z
3023 FOR n=2 TO 5
3024 LET z$(n)="000000": NEXT n
3025 PRINT AT 6,22;z$(1, TO 3);"
*";z$(1,4 TO )
3026 PRINT AT 7,22;"-------";AT
11,22;"-------";AT 13,22;"======
="
3027 LET m$="0123456789"
3028 LET u=0
3029 LET mel=973: GO SUB 900
3030 GO SUB 800
3100 FOR n=4 TO 6: FOR m=3 TO 1
STEP -1
3110 LET p$=STR$ (VAL z$(1,n)*VA
L z$(1,m)+u): IF LEN p$=1 THEN L
ET p$="0"+p$
3111 LET u=VAL p$(1): LET p$=p$(
2)
3120 LET y=4+n: LET x=19+m+n: GO
 SUB 200
3121 IF i$<>p$ THEN GO SUB 500:
GO TO 3120
3122 GO SUB 600
3123 LET z$(n-2,m+n-3)=i$
3130 NEXT m: IF u=0 THEN NEXT n:
 GO TO 3200
3140 LET y=4+n: LET x=19+m+n: GO
 SUB 200
3141 IF i$<>STR$ u THEN GO SUB 5
00: GO TO 3140
3142 GO SUB 600
3143 LET z$(n-2,m+n-3)=i$
3144 LET u=0
3150 NEXT n
3200 FOR n=6 TO 2 STEP -1
3210 LET s=0: FOR m=2 TO 5: LET
s=s+CODE z$(m,n)-48: NEXT m
3211 LET s$=STR$ s: LET u$="0":
IF LEN s$=2 THEN LET u$=s$(1): L
ET s$=s$(2)
3220 LET y=14: LET x=22+n: GO SU
B 200
3221 IF i$<>s$ THEN GO SUB 500:
GO TO 3220
3222 GO SUB 600
3223 IF u$="0" THEN GO TO 3233
3230 LET y=12: LET x=21+n: GO SU
B 200
3231 IF i$<>u$ THEN GO SUB 500:
GO TO 3230
3232 GO SUB 600
3233 LET z$(5,n-1)=u$
3240 NEXT n
3250 LET s$=STR$ (VAL u$+VAL z$(
2,1))
3251 LET y=14: LET x=23: GO SUB
200
3252 IF i$<>s$ THEN GO SUB 500:
GO TO 3251
3253 GO SUB 600
3260 GO SUB 700
3300 RETURN
```

Basic-Listing »Mathe« (Fortsetzung)

```
4000 REM
4001 REM
4002 REM *** Division ***
4003 REM
4010 GO SUB 400: PRINT AT 2,8;"*
** Division ***"
4011 PRINT AT 4,19;"Die Tausende
r";TAB 19;"der    ersten";TAB 19
;"Zahl   werden";TAB 19;"durch
   die";TAB 19;"Zahl    hinter";T
AB 19;"dem   Doppel=";TAB 9;"pun
kt geteilt, und  der";TAB 9;"Gan
zteil des Quotienten";TAB 9;"wir
d notiert.";TAB 9;"Diese Ziffer
 wird dann";TAB 9;"mit der Zahl
hinter dem";TAB 9;"Doppelpunkt m
ultiplizi=";TAB 9;"ert und das P
rodukt no=";TAB 9;"tiert. Nun  f
olgt   die";TAB 9;"Subtraktion u
nd das Ho=";TAB 9;"len der  näch
sten  Zif=";TAB 9;"fer.  Zum  Sc
hluß  wird";TAB 9;"der Rest noti
ert."
4012 LET xa=0: LET dx=6: LET ya=
4: LET dy=18: GO SUB 300: LET xa
=0: LET dx=16: LET ya=4: LET dy=
5: GO SUB 300: PRINT AT 8,1;"
 "

4020 DIM z$(5)
4021 FOR n=1 TO 5: LET z$(n)=CHR
$ INT ((9-(n=5)*RND+49+(n=5))):
NEXT n
4022 PRINT AT 6,2;z$( TO 4);":";
z$(5);"=";"         R"
4023 LET m$="0123456789"
4024 LET r$=z$(1)
4025 LET mel=974: GO SUB 900
4030 GO SUB 800
4040 LET null=0
4100 FOR n=1 TO 4
4110 LET q$=STR$ INT (VAL r$/VAL
 z$(5))
4111 LET p$=STR$ (VAL z$(5)*VAL
q$): IF LEN p$=1 THEN LET p$="0"
+p$
4112 LET r$=STR$ (VAL r$-VAL p$)
+(z$(n+1) AND n<>4)
4113 IF n=1 AND q$="0" THEN LET
null=3: NEXT n
4120 LET x=8+n: LET y=6: GO SUB
200
4121 IF i$<>q$ THEN GO SUB 500:
GO TO 4120
4122 GO SUB 600
4123 IF n=1 THEN GO TO 4140
4130 LET x=n: LET y=3*n+4-null:
GO SUB 200
4131 IF i$<>p$(1) THEN GO SUB 50
0: GO TO 4130
4132 GO SUB 600
4133 PRINT AT y+1,x;"--"
4140 LET x=n+1: LET y=3*n+4-null
: GO SUB 200
4141 IF i$<>p$(2) THEN GO SUB 50
0: GO TO 4140
4142 GO SUB 600
4143 PRINT AT y+1,x;"-"
4150 LET x=n+1: LET y=3*n+6-null
: GO SUB 200
4151 IF i$<>r$(1) THEN GO SUB 50
0: GO TO 4150
4152 GO SUB 600
4153 IF n=4 THEN GO TO 4170
4160 LET x=n+2: GO SUB 200
4161 IF i$<>r$(2) THEN GO SUB 50
0: GO TO 4160
4162 GO SUB 600
4170 NEXT n
4180 LET x=15: LET y=6: GO SUB 2
00
4181 IF i$<>r$ THEN GO SUB 500:
GO TO 4180
4182 GO SUB 600
4190 GO SUB 700
4200 RETURN
5000 REM
5001 REM
5002 REM *** Test ***
5003 REM
5010 GO SUB 400: PRINT AT 2,10;"
*** Test ***"
5011 PRINT '"Sobald Du Deinen Na
men  eingege=ben hast, startet d
er  Test  mitder  Addition,   es
  folgen dannSubtraktion, Multip
likation  undDivision. Bei diese
m  Test kommtes in erster  Linie
  darauf  an,möglichst   wenige
  Fehler   zumachen,    doch  au
ch  die  Zeitspielt eine Rolle."
''"Sei nicht aufgeregt, dieser T
estkann   Dir   schließlich   ni
chtschaden."''"Viel Erfolg!"
5020 INPUT "Gebe bitte Deinen Na
men ein. Be=ende die Eingabe mit
 ENTER."' LINE n$: IF LEN n$>30
THEN GO TO 5020
5030 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
5031 GO SUB 1000
5032 GO SUB 2000
5033 GO SUB 3000
5034 GO SUB 4000
5040 LET t1=65536*PEEK 23674+256
*PEEK 23673+PEEK 23672
5041 LET t2=65536*PEEK 23674+256
*PEEK 23673+PEEK 23672
5042 LET sec=INT ((t1+t2+ABS (t1
-t2))/100)
5043 LET min=INT (sec/60): LET s
ec=sec-60*min
5050 LET p=15-fe+(min<4)-(min>8)
5051 IF p>15 THEN LET p=15
5052 IF p<0 THEN LET p=0
5060 LET z=INT (17/3-p/3+.5)
5070 GO SUB 400: PRINT AT 2,8;"*
** Urkunde ***"''"Hiermit    wir
d    dem    Schüler";
5071 PRINT ''TAB INT (32-LEN n$)
/2;n$
5072 PRINT '"die  Teilnahme    an
  einem  Testder Grundrechenarte
n  bestätigt."
5073 LET xa=0: LET dx=32: LET ya
=11: LET dy=7: GO SUB 300
5074 PRINT AT 13,1;TAB 4-LEN STR
$ fe;fe;" von ";ges;" möglichen
 Fehlern"
5075 PRINT AT 14,2;"benötigte Ze
it:  ";TAB 21-LEN STR$ min;min;"
min";TAB 27-LEN STR$ sec;sec;"se
c"
5076 PRINT AT 15,1;TAB 4-LEN STR
$ p;p;" von 15 möglichen  Punkte
n"
5077 LET xa=0: LET dx=32: LET ya
=19: LET dy=3: GO SUB 300: PRINT
 AT 20,11;"Zensur:  ";z
5078 LET mel=980: GO SUB 900
5079 LET t=1: LET t$="         So
ll diese Urkunde ausgedruckt wer
den, dann gebe 'D' ein; sonst dr
ücke ENTER.      ": LET x=19: LET
y=20: LET m$="D"+CHR$ 13: OVER 1
: GO SUB 200: OVER 0
5080 IF i$="D" THEN PRINT AT 20,
19;z: LPRINT : COPY : LPRINT
5090 RETURN
6000 REM
6001 REM
6002 REM *** Ende des Progs ***
6003 REM
```

Basic-Listing »Mathe« (Schluß)

Variablen-Dump

Im Handbuch des Spectrum wird ausführlich beschrieben, wie die einzelnen Variablen aufgebaut sind. Die Frage ist nur: Was kann man damit machen?

Großcomputer drucken, wenn sie einen Fehler machen, einen ausführlichen Dump aller Variablen. Da ist dann genau aufgelistet, welcher Wert in dem Moment des Fehlers in jeder Variablen steht. Das ist häufig sehr wichtig zu wissen, denn oft lautet die Fehlermeldung schlicht "Overflow in Zeile xxx". Hat man dann eine Liste aller Variablen, so läßt sich der Fehler (der ja oft ganz wo anders steckt) leichter finden.

```
   1 REM Variablenlister
          © 1984 by
          Rainer W. Gerling
   2 CLS : DEF FN p(X)=PEEK X+25
6*PEEK (x+1): DEF FN k(x)=INT (P
EEK x/32): DEF FN l(x)=PEEK x-FN
 k(x)*32+96: DIM a$(14)
   3 LET a=FN p(23627): PRINT "B
eginn der Variablen: ";a: PRINT
: PRINT "Name      Typ     Wert"
   4 LET x=FN k(a): LET l=FN l(a
): PRINT CHR$ l;
   5 IF x=3 OR x=5 THEN GO TO 20
   6 IF x=4 THEN GO TO 40
   7 IF x=7 THEN GO TO 50
   8 IF x=2 OR x=6 THEN GO TO 60
   9 IF PEEK a=128 THEN STOP
  10 GO TO 4
  20 IF x=5 THEN GO TO 30
  21 LET t$="Integer"
  22 IF PEEK (a+1) OR PEEK (a+5)
 THEN LET t$="Real"
  23 PRINT TAB 10;t$;TAB 18;VAL
(CHR$ l)
  24 LET a=a+6
  25 GO TO 9
  30 LET l$=""
  31 LET a=a+1: LET p=PEEK (a)
  32 IF p<128 THEN LET l$=l$+CHR
$ p: GO TO 31
  33 LET l$=l$+CHR$ (p-128)
  34 IF PEEK (a+1) OR PEEK (a+5)
 THEN LET t$="Real"
  35 PRINT l$;TAB 10;t$;TAB 18;V
AL (CHR$ l+l$)
  36 LET a=a+6: GO TO 9
  40 LET p=FN p(a+1): PRINT TAB
10;"Num.Arr";
  41 LET l=PEEK (a+3): FOR i=1 T
O l: PRINT TAB 18;"DIM";i;": ";F
N p(a+2+2*i): NEXT i
  42 LET a=a+p+3: GO TO 9
  50 LET a=a+16: LET p=FN p(a)
  51 PRINT TAB 10;"FORNEXT Zeile
: ";p
  52 PRINT TAB 18;"Stat: ";VAL (
CHR$ l): LET a=a+3: GO TO 9
  60 IF x=6 THEN GO TO 64
  61 LET a$(1 TO 14)=VAL$ (CHR$
l+"$")
  62 LET p=FN p(a+1): PRINT "$";
TAB 10;"String";TAB 18;a$
  63 LET a=a+p+3: GO TO 9
  64 LET p=FN p(a+1): PRINT "$";
TAB 10;"Str.Arr";
  65 LET l=PEEK (a+3): FOR i=1 T
O l: PRINT TAB 18;"DIM";i;": ";F
N p(a+2+2*i): NEXT i
  66 LET a=a+p+3: GO TO 9
  99 SAVE *"m";1;"varlist": VERI
FY *"m";1;"varlist"
```

Listing 1. Das Programm Variablen-Dump

```
Beginn der Variablen: 25278

Name      Typ      Wert
a         Integer  25278
x         Integer  3
l         Integer  108
p         Integer  4
i         FORNEXT  Zeile: 65
                   Stat: 2
xyz       Real     1.2345679E+11
a         Num.Arr  DIM1: 3
                   DIM2: 4
                   DIM3: 5
x$        String   Test
b         Real     2.234
a$        Str.Arr  DIM1: 14
l$        String   yz
t$        String   Real
```

Listing 2. Beispiel-Ausdruck des Programms Variablen-Dump, in dem jeder Variablentyp mindestens einmal vorkommt.

Das Programm in Listing 1 produziert genau solch einen Dump. Wie das geschieht? Das Programm sucht ab VARS nach Variablen. Dann wird der Typ der Variablen bestimmt. Dazu müssen wir die vorderen drei Bits des ersten Bytes einer jeden Variablen untersuchen. Die Funktion k(x) liefert diese drei Bits.

k(x)=2 einfacher String
k(x)=3 einfache Variable
k(x)=4 Zahlen Array
k(x)=5 Variable mit langem Namen
k(x)=6 String Array
k(x)=7 FOR-NEXT Variable

Variablentyp laut Handbuch

Ausgehend von dieser Information wird dann der Variablentyp bestimmt. Die Funktion l(x) liefert die Bytes 0 bis 4 und addiert 96 (60H) dazu. Das liefert den ASCII-Code des Namens (beziehungsweise des ersten Buchstaben des Namens). Damit kann dann der Name (Zeile 4) und der Wert der Variablen (zum Beispiel in Zeile 23) ausgegeben werden.

Außerdem unterscheidet das Programm noch zwischen Integer- und Real-(=Fließpunkt) Zahl. Dazu überprüft es das vorderste und das hinterste Byte der Zahl. Sind beide Null, handelt es sich um eine Integer-Zahl.

Was wird nun alles gelistet? Zuerst einmal Name und Typ der Variablen. Dann wird bei Integer und Real der Wert der Variablen gelistet. Bei einfachen Strings werden aus Platzgründen nur die ersten 14 Zeichen des Strings gelistet. Bei einem Array wird die Dimensionierung angegeben. Der Array a in Listing 2 wurde zum Beispiel als mit DIM a(3,4,5) angelegt. Bei einer FOR-NEXT-Variablen wird Zeilen- und Statement-Nummer des FOR-NEXT Befehls angegeben.

Wie wendet man nun das Programm an? Entweder von Hand mit GOTO 1 oder mit ON ERROR GOTO (Happy-Computer, 11/84, Seite 86). Zum ersten: Man darf den Variablen Dump nie mit RUN starten, da RUN alle Variablen löscht.

Zum ON ERROR GOTO: Wenn das ON ERROR GOTO bei 32500 beginnt, und mit Zeilennummer 1 angelegt wurde, dann ändert man lediglich die Zeile 9 in:

9 IF PEEK a=128 THEN RANDOMIZE USR 32590: STOP

Dann wird der Variablen-Dump, wie bei einem Großrechner, im Falle eines Fehlers und bei Programmende erzeugt.

(R. W. Gerling)

Private Finanzen mit dem Spectrum sicher verwalten

Multikont ist ein brandneues deutsches Programm für den ZX-Spectrum (48 KByte) oder ZX-Spectrum Plus. Das Anwendungsprogramm ist ein rein deutsches Softwarepaket, das es auf Kassette mit Kassetten- und Microdriveversion sowie Demoprogramm und exakter Anleitung gibt. Wenn Sie also mehr als Ihr Taschengeld kontrollieren wollen, lesen Sie bitte weiter.

Der Test zeigt, daß Multikont dank der deutschen Herkunft die oftmals anzutreffenden Schwierigkeiten bei Anwendungen englischer Software umgeht und durch sein gelungenes Konzept viele sinnvolle Anwendungsmöglichkeiten für einen tagtäglichen Gebrauch des Computers im Haushalt aufweist. Multikont dient zur Verwaltung von bis zu zehn Girokonten oder Sparbüchern mit dem Spectrum. Für die Konten können Kontonummer, Bank, Inhaber und so weiter definiert werden. Dann werden im täglichen Gebrauch alle vorhandenen Transaktionen als Buchungssatz mit Datum, Buchungstext und Betrag für eines der bis zu zehn Konten eingegeben. Der große Vorteil dieses insgesamt als sehr gelungen zu bezeichnenden Programmes liegt in den vielfältigen Optionen, die benutzerfreundlich von einem Hauptmenü aus durch einfachen Tastendruck aufgerufen werden können und im Gegensatz zu den üblichen Sammlungen von einzelnen Kontoauszügen eine optimale Übersicht über den privaten Vermögensstand bieten.

Leistung überzeugt

Die einzelnen Optionen sind:

— Kontobewegungen eingeben: hier werden die einzelnen Buchungssätze wie auf einem Kontoauszug eingegeben. Das Programm akzeptiert nur richtige Werte zum Beispiel für Datumseingaben. Als Vorteil ist anzusehen, daß auch bei Transaktionen auf Sparbüchern ein Verwendungszweck als Buchungstext gespeichert werden kann.

— Kontostände abfragen: Ein einfacher Druck auf Taste 1 bis 10 genügt nach dem Abruf dieser Option, um die letzten 16 Buchungen auf dem der Taste entsprechenden Konto abzurufen. Gleichzeitig wird der aktuelle Kontostand ausgegeben. Nun kann man alternativ alle bisher auf diesem Konto eingegebenen Buchungen abrufen, alle Buchungen auf einem angeschlossenen Drucker ausdrucken lassen (dies entspricht einem Kontoauszug für alle vorhandenen Buchungen!) oder einen neuen Stichtag eingeben.

Nach der Eingabe eines neuen Stichtages wird nach kurzer, erträglicher Wartezeit ein Kontoauszug für diesen neuen Stichtag und das betreffende Konto ausgegeben. Somit kann man jederzeit den Kontostand auf seinen Konten zu einem beliebigen vergangenen Zeitpunkt erfahren und ausdrucken lassen.

Gesamtvermögensstand wird berechnet

— Übersicht über alle Kontostände: mit dieser Option werden alle vorhandenen Konten mit Kontostand und der berechnete Gesamtvermögensstand ausgegeben. Zusätzlich erfährt man, wieviele Buchungen bereits gespeichert sind und wieviele das Programm noch verarbeiten kann (maximal 520). Auch hier kann ein neuer Stichtag berechnet werden, eine Hardcopy auf Drucker ist auch möglich.

— Konto eingeben oder löschen: falls Sie ein neues Girokonto/Sparbuch eröffnen oder ein altes mit Saldo 0 auflösen, benötigen Sie diese Option zur Eingabe der spezifischen Daten.

— Datei abspeichern: Nach jeder Eingabe neuer Buchungen muß die aktuelle Datei auf Kassette oder (nur bei der Microdriveversion in Verbindung mit ZX-Interface 1 und Microdrive) Cartridge abgespeichert werden. Dies geschieht mittels entsprechender Option einschließlich eines Verify-Vorgangs.

— Programm beenden.

Gutes Konzept

Wenn das Programm in der Anwendung durch seine vielfältigen, hier noch längst nicht alle aufgezählten Optionen überzeugt, so tut es das erst recht durch sein allgemeines Konzept. Die gelieferte Softwarekassette dient nämlich nur zur Erstellung leerer Multikont-Programme mit Datei. Zunächst wird also das Programm von der Kassette geladen; eine automatische Speicherroutine sichert es dann auf Leerkassette oder Microdrivecartridge ab (und zwar komplett mit leerer Datei). Danach wird das Originalband nicht mehr benötigt (es sei denn, Sie wollen weitere Multikont-Programme zum Beispiel für andere Kalenderjahre anlegen). Alle weiteren Anwendungen erfolgen nur noch von der eigenen Kassette beziehungsweise Cartridge; bei Neueingaben muß nur die Datei mit den Buchungssätzen und nicht das ganze Programm neu abgespeichert werden.

Positives Testergebnis

Der Test zeigt, daß diese Software aus Deutschland sehr sinnvoll angewendet werden kann und neben einigen bekannten englischen Produkten wie Tasword II oder Masterfile als sehr empfehlenswertes Anwendungsprogramm für den Spectrum bezeichnet werden kann. Die knapp 40 Mark für eine Softwarekassette mit speziellen Microdriveoptionen (Ladezeit vom Microdrive beträgt etwa 20 Sekunden) und eine ausführliche deutsche Anleitung stellen sicherlich eine gute Investition dar.

(K. Schober)

Spectrum + Floppy = Profi-System

Mitte letzten Jahres konnten viele Spectrum-Besitzer aufatmen: Endlich gab es auch für diesen Computer ein lauffähiges und datensicheres Floppy-Laufwerk, das die unnötig lange Wartezeit, die beim Laden der Programme vom Kassettenrecorder entsteht, auf ein Minimum verkürzt. Die mitgelieferte Systemsoftware ist zwar gut, aber was bisher fehlte, war ein »Disketten-Doktor«, der alle Informationen von der Diskette liest und Änderungen ermöglicht.

Die Rede ist von der Beta-Disc-Floppy, das über direkten Zugriff verfügt und sich mit einer Ladezeit von etwa 13 Sekunden für ein 48-KByte-Programm auch professionell nutzen läßt.

Doch was nützt die schönste Hardware, wenn die dazugehörende Software fehlt? Und damit taten sich die Software-Hersteller erst einmal schwer. Nur wenige Firmen erkannten die neuen Chancen für den Spectrum und entwickelten floppy-adäquate Anwenderprogramme. Eine davon ist Strecker aus Köln. Drei ihrer Programme sollen hier näher vorgestellt werden.

Multidisc hilft

Ein besonderer Leckerbissen für alle Floppy-Besitzer ist ein Programm, das jedem Benutzer eines Großrechners wohlbekannt sein dürfte und das jetzt auch für den Spectrum zu haben ist: der »Disketten-Doktor« Multidisc.

Das Programm »repariert« softwaremäßig beschädigte Disketten und hilft einmal gemachte Bedienungsfehler zu beseitigen. Wer kennt das Problem nicht? Beim Platzmachen auf der Diskette mit ERASE, löscht man durch einen Bedienungsfehler ein File, das man eigentlich gar nicht löschen wollte. Und schwups, ist die Arbeit vieler Stunden in Sekunden unwiederbringbar entschwunden. Aus lauter Wut möchte man am liebsten Diskette samt Floppy zum Fenster hinauswerfen...

Das braucht man — wie gesagt — jetzt nicht mehr. Es reicht, wenn man Multidisc einlädt und die Diskette »untersucht«.

Dabei erhält man zuerst allgemeine Angaben über die Diskette, wie Titel, Paßword, Anzahl der freien Sektoren, die erste freie Spur, den ersten freien Sektor sowie das Format der Diskette.

Wo steht was?

Als weitere Information zeigt das Programm an, in welchem Track und Sektor sich welches File befindet. Wählt man das entsprechende Menü, erfährt man, um welchen Typ es sich bei dem File handelt (DATA, BASIC oder CODE), außerdem die Programmlänge (bei Basic), die Start- und Ladeadresse und die Filenummer.

Durch einen Sektordump, der alle Operationen wie das Ansehen, Ändern und Schreiben in vorhandene Sektoren umfaßt, ist es möglich, Sektor um Sektor, Track für Track abzufragen und dabei selbst einzelne Bytes zu verändern. Ein Bildschirmeditor vereinfacht dabei die Bedienung.

Die Besonderheit an diesem Utility ist es, einmal ERASEte Files wieder sichtbar zu machen. Da auch diejenige Files, die durch CAT nicht erscheinen, sich immer noch auf der Diskette befinden und bei der Auflistung genannt werden, dürfte das jetzt keine Schwierigkeiten mehr bereiten. Man erkennt die ERASEten Files an einem Copyright-Zeichen an erster Stelle des Filenamens. Durch einfaches Ändern dieses Namens, wobei selbst diese Operation menügesteuert ist, erscheint das File wieder im Katalog der Diskette.

Den einzigen Fehler, den man nicht machen darf, wenn man durch ERASE auf der Diskette Platz schafft, ist, MOVE zu betätigen, denn dann sind wirklich alle ERASEten Files für immer verloren. Dann nützt es auch nichts mehr, die ausführliche deutsche Bedienungsanleitung, die jedem Programm beigefügt ist, genauestens zu lesen oder die sechs Menüs dieses Programms, die es so bedienungsfreundlich machen, nach einer Möglichkeit durchzustöbern. Die Diskette bleibt, wie ihr Katalog aussieht: leer.

Programmerweiterungen gewünscht?

Die beiden anderen Programme, die hier vorgestellt werden sollen, sind im Grunde Floppy-adäquate Erweiterungen zu schon bestehenden Programmen: Multitas zum Tasword und Multipas zum Highsoft Pascal. Beide laufen ohne die Orginal-Programme nicht, bieten dafür eine Vielzahl von neuen Anwendungen. So ist es zum Beispiel mit Multipas möglich, DOS-Befehle aus dem Compiler aufzurufen und die Include-Funktion auf Diskette ausführen zu lassen. Beim Multitas wurde die Option für die Blockverschiebung stark verbessert und die Funktion, nach bestimmten Worten im Text zu suchen, wesentlich beschleunigt.

Hinzu kommt noch, daß das Tasword mit Hilfe von Multitas völlig eingedeutscht wird, sogar die Helpseiten. Was die Umlaute anbetrifft, so liegen sie jetzt da, wo sie liegen sollen: das ä auf dem a, das ö auf dem o und so weiter (auch im 32-Zeichen-Modus).

Summasummarum, betrachtet man diese bedienungs- und anwenderfreundlichen Programme, so gehen Beta-Disc-Floppy-Besitzer rosigen Zeiten entgegen... Weitere Anwenderprogramme dieser Qualität sind bereits in Vorbereitung.

(Karina Krawczyk)

Goldene Diskette für Lernprogramm auf dem Sinclair ZX Spectrum

Im Wettbewerb um die Goldene Diskette für hervorragende Programmierleistungen kam auch ein Programm für den Spectrum zu Ehren. Geschrieben wurde das Geografie-Lernprogramm Geomat von dem siebzehnjährigen Schüler Olaf Hartwig, der zusammen mit fünf anderen Preisträgern auf der diesjährigen Hannover-Messe ausgezeichnet wurde.

Wie viele Tage ich gebraucht habe, um Geomat zu schreiben?« Der 17 Jahre alte Preisträger im Wettbewerb um die Goldene Diskette 1985, Olaf Hartwig, überlegt keinen Moment: »14 Tage und Nächte, dann stand das Programm«.

Olaf, Gymnasiast aus Reinhardshagen an der Ostsee, zählt zu den sechs Jugendlichen, die für hervorragende Leistungen vom Bundesministerium für Forschung und Technologie ausgezeichnet worden sind. Zusammen mit fünf weiteren Preisträgern im Alter zwischen 12 und 19 Jahren erhielt er im Informationszentrum Jugend und Technik der Hannover-Messe einen Preis für das beste Lernprogramm — Geomat für den Spectrum.

Begonnen hat alles vor sieben Jahren. Auf Initiative des Physiklehrers am Klaus Harms Gymnasium in Kappeln lernte Olaf das Programmieren. Als er sich vor zwei Jahren einen Taschenrechner für die Schule kaufte, fing er an, in Basic zu programmieren. Bald fand er heraus, daß er kleinere, ganz einfache Programme an Freaks weiterverkaufen und damit sein Taschengeld aufbessern konnte. Von diesen Einnahmen kaufte er sich einige Monate später seinen ersten richtigen Computer, den Sinclair ZX Spectrum.

Nun stand den Programmierbemühungen von Olaf nichts mehr im Wege. Er entwickelte für den Spectrum Arcade und Adventure Games — meist mit 3D-Grafik — neuartige Tools, Utilities, Grafik — und Sound-Erweiterungen.

Aber auch bereits kommerzielle Anwendungen. Da gibt es beispielsweise ein Trennungsprogramm, mit dem man laut Olaf »die komplizierten Duden-Trennungsregeln ein für allemal vergessen kann«. Oder der Finance Manager, der bis zu fünf verschiedene Konten parallel führt und einen ständigen Überblick über das gesamte Guthaben und aktuelle Einzelkontenstände gibt.

Die rund 40 Programme und Erweiterungen stellte der Gymnasiast in einem Katalog zusammen, meldete ein Gewerbe an und inserierte in Computerzeitschriften.

Einer der Sieger im Wettbewerb um die Goldene Diskette: Olaf Hartwig (17) aus Reinhardshagen (Programm »Geomat«).
In Vertretung des Bundesministers für Forschung und Technologie, Dr. Riesenhuber, überreicht Dr. Donth die Auszeichnung. Außerdem erhält Olaf einen Computer der Firma Compaq.

Gewerbliche Nutzung

Die Taschengeld-Einnahmen stiegen, halten sich aber in Grenzen, weil Olaf seine Zeit noch auf vielfältige andere Interessen verteilt. Er spielt Gitarre, absolviert sein tägliches Jogging-Programm, fotografiert und surft den Sommer über auf der nahen Ostsee. Gerne hört er auch englische Sender über Kurzwellen-Empfang.

Und dennoch: im stillen Kämmerlein arbeitet er bereits an einem Buch über künstliche Intelligenzforschung. Olaf Hartwig: »Es geht darum, wie man den Computern das Denken beibringen kann.«

Und mit einigem Stolz weist er auf seinen zweiten Platz hin, den er im Landeswettbewerb Schleswig Holstein »Jugend forscht« für ein statistisches Analyse-Software-Paket erhalten hat. Auf dem Spectrum entwickelte er die Spracheingabe. Vom VDI und Landeskultusminister erhielt er zusätzlich den Sonderpreis für die originellste und kreativste Arbeit.

Für die Zukunft hat der vielseitige Schüler bereits feste Pläne. Im Frühjahr 1986 Abitur — Leistungsfächer in der Schule sind Mathematik und Chemie — dann ein Informatikstudium. Ach ja, und dann natürlich Amerika, ein Aufenthalt im Mutterland der Bits und Bytes. (mk)

Geomat für den Spectrum

Das für den Spectrum geschriebene Programm Geomat ist ein Geographie-Lernprogramm, das nicht nur Wissen vermittelt und abfragt, sondern auch selbst fähig ist, zu lernen. Zunächst einmal dient das insgesamt dreiteilige Programm als elektronisches Geographie-Lexikon. Einzelne Länder mit wichtigsten Daten werden auf einer farbigen Weltkarte dargestellt.

Der zweite Teil besteht aus einem Wissenstest, der dritte Teil schließlich ermöglicht es, das Programm geografische Tatbestände erraten zu lassen. Dies geschieht über Suchbäume, die zum Teil der Entwicklung von Expertensystemen entlehnt sind. Wenn das Programm einen Begriff nich errät, fragt es selbst danach und nach Kriterien, unter denen es den Begriff beim nächsten Ablauf in sein Wissen einbauen und abfragen kann. Das Programm ist sehr gut dokumentiert und nutzt die Möglichkeiten des Computers voll aus.

Daten-Verbindungen

Wenn Sie mehr aus Ihrem Spectrum herausholen wollen, müssen Sie wissen, wie dieser an welcher Schnittstelle Daten zur Verfügung stellt. Als Lohn des Fleißes locken neue Verbindungen.

Der Spectrum kann in Verbindung mit dem Interface 1 auf drei Arten mit Peripheriegeräten oder auch mit anderen Computern kommunizieren: über Kassettenrecorder-, RS232- und Netzwerkschnittstelle. Am einfachsten zu realisieren ist die Verbindung über die RS232-Schnittstelle. Diese erlaubt sowohl das Senden wie auch das Empfangen von Daten. Außerdem ist die serielle Schnittstelle fast genormt und die dafür nötige Softwareanpassung kann auch ein Anfänger in Basic erstellen. Schwieriger ist die ganze Sache mit dem Networkanschluß. Der Datentransfer erfolgt hier mit 16 KBaud am schnellsten, aber wesentlich komplizierter. Ohne Maschinensprachekenntnisse ist nichts zu machen.

Nur für echte Freaks ist das Kassetteninterface zur Datenübertragung zu empfehlen. Sowohl von der Hardware, wie auch der Software müßen sehr gute Kenntnisse vorliegen.

V.24=seriell

Da viele Computer über eine serielle Schnittstelle verfügen, ist das Übertragen von Daten und Programmen mit dem seriellen Ausgang des Interface 1 häufig anwendbar. Nach dem Starten des Spectrum ist bei dem Interface 1 immer die Übertragungsgeschwindigkeit von 9600 Baud eingestellt.

Insgesamt stehen alternativ neun Übertragungsgeschwindigkeiten zur Verfügung 50, 110, 300, 600, 1200, 2400, 4800, 9600 und 19200 Baud. Davon abweichende Baudraten werden in der Praxis nur selten verwendet, können aber über Maschinencodeprogramme erzeugt werden. Wählt man aus dem Basic heraus eine Baudrate die zwischen den normalen Werten liegt, so reduziert das Betriebssystem des Spectrum den Wert auf den nächsten darunterliegenden. Aus FORMAT "t";150 wird 110 Baud. Da zwischen den Bytes kurze Pausen nötig sind, ist die tatsächliche Geschwindigkeit unter der angegebenen.

Mit der Befehlsequenz »FORMAT"b";300« setzt man den "b"-Mode, sowie die Baudrate. Durch die Gerätekennung "b" oder "t" wird zwischen dem 7 Bit-Textmode oder dem 8 Bit-Datenmode gewählt.

Die häufigste Anwendung findet der "t"-Mode zum Ansteuern von Druckern. Dieser 7 Bit-Mode ignoriert alle Steuerzeichen außer CR, (aus dem CR und LF wird) der den Zeilenvorschub bewirkt. Grafikzeichen setzt er in Fragezeichen um und für die Basic-Tokens (den Schlüsselworten auf den Tasten) werden die entsprechende Worte ausgegeben. Der "t"-Mode erleichtert es, Programmlistings des Spectrum auf andere Computer zu übertragen und dort zu bearbeiten. Dazu benötigt man eine Übertragungssoftware, wie das Programm »MOVE IT« für den IBM-PC oder »Async« für den Apricot PC. Die Übertragung erfolgt mit dem Kommando LLIST. Das komplette Programmlisting läßt sich anschließend mit einem Textverarbeitungsprogramm bearbeiten. Ein Listing wieder in den Spectrum zu übertragen und ablaufen zu lassen ist nicht möglich, da die Systemvariablen und Ablaufvariablen fehlen. Aber Listings, die ja auf eine Zeilenbreite von 32 Zeichen begrenzt sind, können im 80-Zeichen-Mode einer Textverarbeitung sehr gut mit Kommentaren versehen und ausgedruckt werden. Das Senden von Dateien erfolgt nach dem gleichen Prinzip mit »PRINT # 3;a$«.

»b« gleich Binär

Um Datenblöcke, Speicherblöcke oder Programme zu senden oder zu empfangen, verwendet man den "b"-Mode. Damit überträgt man Werte zischen 0 und 255. Das ermöglicht es, Programme ablauffähig zu übertragen. Es muß nur der gesamte Speicherbereich von den Systemvariablen bis zum Programmende übertragen und auch wieder zurückgeladen werden. Diese Werte lassen sich einfach ermitteln. »PRINT PEEK 23641 + 256*PEEK 23642« ergibt den Wert der Systemvariablen E_LINE, der Endadresse des Programms und der Variablen. Der Anfang der Systemvariablen liegt immer bei 23552. Von dem aus

der Variablen E_LINE ermittelten Wert zieht man 23552 ab und hat die Länge des Programms. Um das Programm zu übertragen, öffnet man den Kanal 3 mit dem Befehl »OPEN#3;"b"«. Nun setzt man den empfangenden Computer in den WAIT-Mode, also empfangsbereit, und gibt »SAVE*"b"CODE 23552, (Länge des Programms)« ein.

Der Programmname wird vom empfangenden Gerät bestimmt. Diese Befehlsfolge kann auch in ein Programmlisting eingebunden sein. Empfängt der Spectrum dieses Programm, geht der Programmlauf beim folgenden Befehl weiter, das Programm ist selbststartend! Zu beachten ist dabei eines: Zwischen RAMTOP (PRINT PEEK 23730 + 256*PEEK 23731) und STK_END (PRINT PEEK 23653 + 256*PEEK 23654) müssen mindestens 11 Byte frei sein für die temporären Variablen. Diese enthalten die Startadresse, die Kennung für 7 oder 8 Bit-Mode, Ein- und Ausgabeadresse sowie die Zahl der belegten Bytes. Das Betriebssystem gestattet es, bis zu 16 Kanäle gleichzeitig zu öffnen. Wobei die Kanäle 0 bis 3 immer vom System belegt sind. Der Druckerkanal kann ohne Probleme benutzt werden. Auf diesen sprechen auch die Druckbefehle LLIST und LPRINT an. Hilfreich ist das öffnen von zwei Kanälen für das Aussenden von Steuercodes für den Drucker und dem Senden von druckbaren Zeichen. Für das erstere werden 8 Bit benötigt und für das zweite 7 Bit. Auch beim Datensenden kann das nötig sein, da das Async-Programm für die Apricots zum Beispiel Character 26 als Endeerkennung benötigt.

Manche Computer senden auch nach der Enderkennung noch Daten. Das kann ein Grund für Fehler bei der Datenübertragung sein. Empfehlenswert ist es dann, eine niedrigere Baudrate zu wählen, um die Fehlerquote zu senken. Bei der Übertragungsgeschwindigkeit von 19200 Baud ist die theoretische Bitlänge 182,3 Takte (1 Takt = 1/3500000 Sekunde). Die tatsächliche Zeit ist aber 177 Taktzyklen. Das bedeutet einen Fehler von zirka 3 Prozent. Je kleiner die Baudrate ist, desto geringer ist auch der Fehler. Leider ist es dennoch oftmals erforderlich, eine Baudstufe niedriger als angenommen zu wählen.

RS232-Verkabelung

Die RS232-Hardware besteht aus sechs Leitungen. Zwei werden zum Senden benutzt, zwei beim Empfang, einer als Masseleitung und der letzte für die Versorgungsspannung. Die erste Leitung wird zum Senden benutzt und heißt RXDATA oder RECEIVE DATA. Die zweite Leitung nennt man DTR oder DATA TERMINAL READY, sie wird vom empfangenden Gerät benutzt, das hier mitteilt, ob es bereit ist Daten zu empfangen. Das Senden eines Byte geht folgendermaßen vor sich: DTR wird auf Highpegel gelegt und erst dann das Datenbyte übertragen. Für jedes Zeichen werden 11 Bits gesendet: Ein Startbit, acht Datenbits und zwei Stopbits. Die Bezeichnung der dritten Verbindung lautet TXDATA oder TRANSMIT DATA. Das Gegenstück zur zweiten Leitung nennt sich CTS oder CLEAR TO SEND. Das Empfangen läuft wie folgt ab: In der Betriebssystemvariable SER FL wird geprüft, ob der Spectrum empfangsbereit ist. Das wird durch die Abfrage auf ungleich Null gemacht. Anschließend setzt er CTS auf Highpegel, wartet auf das Startbit der TXDATA-Leitung, ließt anschließend die acht Datenbits und speichert dieses Byte ab. Nun setzt er CTS auf Lowpegel. Jetzt dürfte kein Zeichen mehr gesendet werden, einige Computer tun das leider doch. Deshalb wird TXDATA weiterhin überprüft. Kommt ein weiteres Zeichen, erfolgt wieder: Lesen der Datenbits und speichern in das höherwertige Byte von SER FL, das niederwertige wird auf 1 gesetzt.

Ins Netz gegangen

Das lokale Netzwerk stellt eine schnelle und unkomplizierte Möglichkeit dar, zwischen Spectrum oder/und Sinclair QL einen Datentransfer durchzuführen. Bis zu 64 Sinclair Computer können zu einem Netzwerk zusammengeschlossen werden. Schaltet man den Spectrum (mit Interface 1) ein, so setzt das Betriebssystem des Interface 1 die Stationsnummer auf 1. Werden nicht mehr als zwei Geräte verbunden, braucht die Stationsnummer nicht geändert werden. Bei mehr als zwei Computern können Daten an eine zu bestimmende Station gesendet werden, die auch nur diese empfangen kann. Die Stationsnummern gehen von 1 bis 64. Die Nummer 0 dient zum Senden an alle Computer im Netz. Jeder Computer, der auf Empfang geschaltet ist, empfängt diese Daten. Das Ändern der Stationsnummer erfolgt mit dem FORMAT-Befehl. »FORMAT "n",2« setzt die Stationsnummer 2. Ein Computer darf auch mehrere Stationsnummern haben, etwa eine zum Senden von Daten und eine zum Empfangen von Programmen.

Das Übertragen geht mit den üblichen Befehlen vor sich: »SAVE*"n",2« überträgt an Station 2 ein Programm, »LOAD*"n",0« lädt das Programm. Programmnamen sind nicht notwendig. Eine Sendung an alle Computer im Netzwerk, Empfängerstation 0, überträgt der Spectrum zirka vier mal langsamer als eine direkt adressierte Sendung. Er wartet auch nicht auf eine Bereit-zu-Empfangen-Mitteilung, sondern sendet einfach die Daten. Wer zu diesem Zeitpunkt noch nicht bereit war, geht leider leer aus. Die üblichen Anhängsel wie LINE, SCREEN$, DATA und CODE können weiterhin verwendet werden.

Mit vier Befehlen alles im Griff

Um Daten direkt zu Senden und zu Empfangen stehen die Befehle OPEN#, INPUT#, INKEY$# und CLOSE# zur Verfügung.

»OPEN#4;"n";2« öffnet den Kanal 4 und weist ihn dem Spectrum mit Stationsnummer 2 zu. Station 2 muß auch einen Kanal für die Eingabe öffnen, »OPEN#5;"n";1«. Um Daten zu Senden verwendet man die Kommandos PRINT# und LIST#. LIST# überträgt das Programmlisting, PRINT# überträgt Zeichen und Variable. Ablauffähig ist ein übertragenes Programm natürlich nur auf einem Spectrum, der Sinclair QL verwendet Superbasic. Diese Programmiersprache ist dem Basic-Dialekt des Spectrum zwar sehr ähnlich, aber Programme laufen leider nicht ab. Benutzt man zum Einlesen INKEY$#, so muß man auf eine Besonderheit achten: Der Spectrum gibt, wenn kein Zeichen von der Sendestation kommt, ein CH$(0) aus. Dieses Problem läßt sich sehr einfach umgehen, indem man auf CHR$(0) abfragt und nur ungleich CHR$(0) anzeigen läßt.

Zwei Leitungen reichen

Das Netzwerk arbeitet nur mit zwei Leitungen. Eine Leitung davon ist Masse, die andere ist die Signalleitung; aktiv liegt sie auf +5 Volt, inaktiv auf Masse.

Benutzt man das Netzwerk, so sendet es als erstes einen Startimpuls und die Stationsnummer aus. Diese Daten werden langsamer übertra-

gen, als die Übermittlung ansonsten vor sich geht. Die Übertragung der Stationsnummer geht in umgekehrter Reihenfolge vor sich, also die Bit 7 zuerst, Bit 0 zum Schluß. Jedes Bit wird überprüft, ob es vom Netzwerk angenommen wurde. Ist das Netzwerk gerade belegt, so wird der ganze Vorgang wiederholt. Nachdem diese Vorabinformationen gesendet und angenommen wurden, muß der Vorspann ausgesandt werden. Der Vorspann enthält:

Vorspann-Inhalt

Die Nummer der Gegenstation, die eigene Stationsnummer, eine Blocknummer (0-65535), Ende-der-Datei Kennung (1 = Daten, 0 = Ende), Länge des Puffers (255 beim Senden, 0 beim Empfang), Prüfbyte der Daten (8-Bit Quersumme), Prüfbyte des Vorspannes (8-Bit Quersumme der 7 Vorspannbytes),
Nur beim Empfangen:
Speicheradresse des letzten Bytes aus dem Puffer, Anzahl der verbleibenden Bytes im Speicher.

Das Aussenden des Vorspannes erledigt die OUTPAK-Routine in folgender Abfolge:
ein aktiver Vorimpuls,
ein inaktiver Startimpuls,
ein aktiver oder inaktiver Impuls für jedes Bit,
ein aktiver Stopimpuls,
Netzwerk inaktiv schalten.

Ist das Aussenden beendet, prüft die Routine den Inhalt der Systemvariablen NCIRIS, die Stationsnummer des Empfängers. Steht hier als Empfänger 0, wartet das System nicht, sondern sendet sofort weiter. Ansonsten sendet die OUTPAK-Routine der gezielt angesprochenen Station den Wert 1 als Quittung. Kommt innerhalb einer bestimmten Zeit nicht die Quittung von der Empfangsstation, so sendet der Spectrum die Belegungsprüfung und den Vorspann noch einmal und wartet wieder auf die Antwort. Angezeigt wird das durch die gleichbleibende Bildrandfarbe.

Viele Prüfungen

Wenn der Vorspann empfangen wurde, laufen im System umfangreiche Prüfungen ab: Ist der Vorspann acht Byte lang? Ist die Quersumme der ersten sieben Bytes richtig (Achtes Byte)? Ist die Blocknummer (NTNUMB) richtig? Wenn an Station 0 gesendet wurde muß NTDEST und NCIRIS gleich sein, wenn nicht dann NTDEST und NCSELF. Ist die Nummer der Sendestation richtig (also NTSRCE und NCIRIS gleich)? Bei einer Sendung an alle Netzwerkteilnehmer erfolgt keine Rückmeldung. Wird an eine bestimmte Station gesendet, muß eine Meldung erfolgen.

Kommt die richtige Antwort, überträgt die OUTPAK-Routine den ersten Datenblock. Das E-Register enthält die Menge der Bytes (NCB), das HL-Register die Anfangsadresse des Puffers.

Die Datenblöcke werden von 0-65535 numeriert. Öffnet man einen Netzwerkkanal, setzt das den ersten Block auf 0. Der letzte Block enthält nur die Ende-der-Datei-Kennung. Versuchen mehrere Computer sich gleichzeitig im Netzwerk anzumelden, so wird der mit der niedrigeren Stationsnummer bevorzugt.

Zeit im Griff

Hier einige technische Daten über den Zeitablauf beim Netzwerk: Bei der Vorbelegung prüft der Spectrum etwa 10500 Takte (3 ms) des Netzwerks auf aktive Impulse. Wird keiner festgestellt, gilt das Netzwerk als nicht belegt. Der Vorimpuls im Belegungsversuch startet 22 Takte nach der Freiprüfung. Dieser dauert 181 Takte, genau wie die Übertragung der 8 Bits der Stationsnummer. Nach 136 Takten werden die Impulse zur Überprüfung zurückgelesen. Nun gilt das Netzwerk als belegt. Nach einer Zeit von 271 Takten werden die acht Bytes des Vorspannes übertragen. Das Übertragen des Vorspannes läuft nach folgenden Muster ab: 98 Takte dauert der Vorimpuls, 40 Takte für den Startimpuls, 40 Takte für jedes der acht Datenbits, 145 Takte dauert der Stopimpuls, zirka alle 7000 Zyklen (2 ms) prüft der Spectrum den Freizustand des Netzwerkes. Anschließend untersucht er es alle 55 Takte auf einen aktiven Impuls. So wird ein Vorimpuls bei einem Belegungsversuch immer in seinem ersten Drittel erkannt. Anschließend ruft er, um zu empfangen, die INPAK-Routine auf. Sie empfängt diese acht Byte des Vorspanns. Der Einsprung erfolgt in der inaktiven Phase von 271 Takten zwischen Belegungsanforderung und Vorspann. Anschließend untersucht das System alle 35 Taktzyklen des Netzwerks, um einen Vorimpuls zu finden. Nun synchronisiert sich die Hardware auf die abfallende Flanke des Signals beim Beginn des ersten Startimpulses. Die Unsicherheit bei der Flankenerkennung wird dadurch auf ein Minimum reduziert.

Der richtige Takt

Die Datenbits werden jetzt mit 40 Takten pro Bit eingelesen. Dabei ist eine hardwarebedingte Abweichung von 5 Prozent bei der Taktfrequenz möglich, nimmt aber keinen negativen Einfluß auf die Übertragung. Wenn alle acht Vorspannbytes empfangen wurden, müssen sie geprüft werden. Ist es eine Sendung an Station 0 wird keine Quittung erteilt. Innerhalb der nächsten 600 Takte wird der Spectrum dann den Datenblock empfangen. Ist die Sendung an eine Station adressiert, muß die Empfangsbestätigung innerhalb von 9000 Takten erfolgen. Ansonsten wiederholt der Sendecomputer den Block. Enthält ein Block ein Byte, dauert das Senden 544 Takte, bei 255 Byte sind es etwa 128000 Takte (37 ms).

Daten nur für Recorder

Die Kassettenrecorderschnittstelle zur Datenkommunikation zu benutzen, ist nicht zu empfehlen, da sich einige Fehler einschleichen können. Wird ein Programm auf Kassette geSAVEt, so erfolgt zuerst ein Signal von rund 5 Sekunden. Der Spectrum sendet keine Daten, sondern gibt der Aufnahmeautomatik Zeit, sich auf die richtige Lautstärke einzupegeln. Anschließend überträgt er den Vorspann oder Header. Dieser enthält den Programmnamen. Es werden immer die vollen 10 Byte des Namens übertragen. Das nächste Byte enthält eine Kennung, die besagt ob es sich um ein Programm, einen Screen (also Bildschirminhalt), einen Datenblock oder einen DATA-Bereich handelt.

Mit Köpfchen

Anschließend erfolgt die Übertragung der Felder für die Adresse, ab der gespeichert wurde und die Länge des Programmes und ab welcher Zeilennummer das Programm selbst starten soll. Steht in diesem Feld 0, startet das Programm nach dem Einlesen in den Rechner nicht selbst. Bei einer Programmübertragung steht die Adresse, ab der das Programm eingelesen wird, fest. Wird ein Bildschirminhalt mit der Zusatzfunktion SCREEN$ gespeichert, so stehen nicht nur Anfang, sondern auch die Länge fest (6912 Byte => 24*256 Byte für den Graphik- und 768 Byte für den Farbspeicher). Ansonsten müssen diese Werte ermittelt werden. Die Übertragung erfolgt Bitweise. Wird der

Bitwert 0 gesendet, entspricht die Übertragungszeit (der Impuls) 855 Taktzyklen, enthält der Bit Wertigkeit 1, so dauert der Impuls 1710-Zyklen, also genau die doppelte Zeit. Nach dem Sendevorgang eines Blocks wird ein Kontrollbyte übertragen, daß auf einer OR-Abfrage basiert. Dieses Kontrollbyte war beim ZX81 verantwortlich, wenn der Rechner das ganze Programm gelesen und dann gelöscht hatte. Er führte diese Abfrage einmal durch, und bei negativen Ergebnis löschte er den gesamten Speicherbereich durch Restart 0.

Der Spectrum macht das nicht, das heißt ein Programm, daß beim verifizieren eine Fehlermeldung verursacht hat, kann trotzdem wieder eingelesen werden. Ob es ablauffähig ist, muß sich dann erst zeigen.

Die Recorderschnittstelle ist durch die Hardware nur zum Senden und Empfangen von ganzen Blöcken geeignet und nicht zum Handshakebetrieb, wie die RS232- oder Netzwerkschnittstelle.

Dadurch, daß die Stromabgabe bei zirka 0.2 Volt und die Stromaufnahme bei zirka 1 Volt liegt, müßte man mit zusätzlicher Hardware arbeiten. Die Toleranz liegt bei etwa 10 Prozent. Nach spätestens 1100 Zyklen muß der Spectrum die nächste Flanke empfangen, wenn es zu keinem Fehler kommen soll.

(Horst Brandl)

Microdrive-Express

Möchte man die Ladezeiten beim Commodore-Laufwerk verkürzen, ist dies nur mit reichlichem finanziellen Aufwand zu erreichen. Bei den Spectrum-Microdrives bekommt man diese Möglichkeit frei Haus und schützt sich auch noch gegen Datenverlust.

Über die Sinclair-Microdrives ist viel geschrieben worden, und nicht alles davon war übermäßig positiv. Diesem Speichermedium haftet der Ruf an, Daten und Programme möge man ihnen besser nicht anvertrauen, da es leicht zum Verlust führen könne. Sicher wird diese Meinung nicht von allen Benutzern geteilt und bei mir kommt nach einjähriger intensiver Benutzung der Microdrives ein wenig der Verdacht auf, daß dieses Gerücht von Fall zu Fall ebenso überprüft werden sollte, wie die »Tatsache«, daß Käse dumm mache.

Auch die Sinclair bemüht sich, das lädierte Image der Microdrives ein wenig aufzupolieren. So verkündet man, daß mittlerweile die Verläßigkeit der Microdrives so groß ist, daß bei bis 50000-fachem Zugriff kein Datenverlust beobachtet wurde.

Wie immer die Sache aussieht, im Prinzip sind die Microdrives eine preiswerte Alternative zu Diskettenstationen. Mittlerweile erhält man das Interface 1 und zwei Laufwerke für rund 600 Mark, was etwa dem Preis eines Commodore-Laufwerkes entspricht. Die Speicherkapazität beträgt in beiden Fällen etwa 170 KByte, jedoch erhält man bei Sinclair eine RS232-Schnittstelle kostenlos dazu. Darüber hinaus sind die Lade- und Schreibzeiten bei Microdrives etwa 10mal kürzer als beim Commodore-Laufwerk.

SEC	LEN	T	NAME EXPRESS 41.0
127	6.5K	B	TASWORD
120	12.0K	C	tasword
99	6.5K	B	TASWORD
70	12.0K	C	tasword
41	0.5K	B	run
37	0.5K	B	run
33	0.5K	B	run

3x run und 2x Tasword auf einem Cartridge

Weiterhin ist es natürlich sehr angenehm, zwei Laufwerke zu haben, wenn man an praktische Arbeit wie Kopieren denkt. Für jeweils weitere 200 Mark lassen sich bis zu insgesamt 8 Laufwerke an den Spectrum anschließen, was zu einem Zugriff auf über 640 KByte führen würde.

Seit März sind nun die Preise für die Microdrive-Cartridges drastisch gesenkt worden. Damit kommt eine denkbar einfache Möglichkeit zum Tragen, die Zugriffszeit auf die Microdrives zu verkürzen und dabei auch noch etwas für die Datensicherheit zu tun. Einziger Nachteil dieses »Microdrive-Express« ist, daß die auf diese Art gesicherten Programme und Daten etwa doppelt so viel Platz auf der Kassette in Anspruch nehmen.

Da man aber in der Regel die Kapazität gar nicht voll ausnutzt und die Methode nur für die am häufigsten benutzten Files beziehungsweise wichtigsten Programme anwenden muß, ist dies wohl kaum ein entscheidender Nachteil.

Der Trick ist ganz einfach, man sichert das entsprechende Programm einfach zweimal (oder noch häufiger) auf das Cartridge. Nun läßt der Spectrum das nicht so einfach zu, nach dem ersten Sichern erscheint die Meldung »Writing to a read file«. Dies läßt sich jedoch einfach umgehen. In der Speicherstelle 23791 steht eine Systemvariable, die die Anzahl der auf Microdrive durchgeführten Kopien angibt, also in der Regel eine 1. Geben wir vor dem Sichern ein »POKE 23791,2«, so werden tatsächlich zwei Kopien angefertigt. Natürlich können auch nach Belieben drei, vier oder mehr Kopien angefertigt werden.

Lädt man nun ein gesichertes Programm wieder, verkürzt sich die Zugriffszeit in der Tat ganz erheblich, weiterhin sind die Daten mehrmals auf der Cassette und sind so gegen Verlust besser geschützt.

(P.C. Bosetti)

PEEKs und POKEs

Tips, die das Leben eines Programmierers erleichtern helfen und (leider) nicht im Handbuch stehen.

Umgehen der Meldung »Start tape, then press anny key.« 1000 FOR a=1 TO 5: PRINT a: SAVE n$(a): PAUSE 150: POKE 23736,181: NEXT a

Listschutz:
Wenn das Basicprogramm als Code abgespeichert wurde, dann gibt man als erste Zeile ein: 1 LOAD ""CODE 24000:POKE 23635,139:POKE 23636,94: LIST

Fehlerrücksprungadresse ändern: »POKE 23614,100« in die erste Programmzeile bringen. Das Programm muß mit "LINE" gesaved werden.

Basic-Programmstartadresse:
PRINT PEEK 23635+256*PEEK 23636 (normalerweise 23755)

Erste Basiczeile auf Null setzen:
LET a=PEEK 23637+256*PEEK 23638:POKE a,0:POKE a+1,0
POKE 23756,x: Setzt erste Zeile im Programm auf Zeilennummer x. Wenn x=0, kann nicht mehr (so leicht) gelöscht werden. Der POKE-Befehl kann natürlich dazu benutzt werden, das Ganze rückgängig zu machen.

RAMTOP:
PRINT PEEK 23730+256*PEEK 23731 (maximal: 32767 bei 16 KByte beziehungsweise 65535 bei 48 KByte)

Freier Speicherplatz:
PRINT;"noch";PEEK 23730+256*PEEK23731-PEEK23653-256*PEEK 23654;"Bytes"
Es geht auch einfacher: PRINT 65535 — USR 7962

Tastatuklick:
POKE 23609, x=Länge des Tones; max. 255

Warnton:
POKE 23608, x=verändert die Länge des Warntones, wenn Programmzeile länger als Bildschirmseite

Repeat:
POKE 23562, x=Beschleunigung in 1/50 sec. x= 5 normale Geschwindigkeit

Repeat-Wartezeit: Wartezeit, bis Repeatfunktion einsetzt POKE 23561, x=Zeit; beim Einschalten ist x=35
Zeitzähler auf Null setzen:
POKE 23672,0:POKE 23673,0:POKE 23674,0

Automatischer Scroll:
POKE 23692,x (x > 1) oder Randomize USR 3280 oder Randomize USR 3582 oder Zeile INPUT INKEY$ einfügen

Editierzeile Attribute ändern: Farbe, Bright, und so weiter.
POKE 23624, BIN XXXXXXXX
X = 1 oder 0 probieren
0 — schwarz
11111111 — hell weiß

Um 256 kleinere Anfangsadresse des Zeichensatzes im ROM:
POKE 23606,x

Änderung des Zeichensatzes durch Ablegen im RAM durch POKE 23606; Adresse im RAM-256; dann wie UDG verfahren, jedoch vorab CLEAR... (Adresse -1)
Hier wird als Beispiel meistens POKE 23606,8 angegeben. Das soll einen dem russischen ähnlichen Zeichensatz ergeben.
Folgendes Programm zeigt die Veränderung deutlich:
1 FOR b=0 TO 255: POKE 23606,b
2 FOR n=32 TO 255: PRINT CHR$ n;:NEXT n:PRINT AT 21,0;"23606,", b: PAUSE 50: CLS: NEXT b: STOP
POKE 23607,10: Zeichensatz wird Punktehaufen
POKE 23607,60: Zeichensatz wird wieder sichtbar

Fertige Routinen im ROM

RANDOMIZE USR 3435 = CLS
RANDOMIZE USR 3756 = COPY
RANDOMIZE USR 4535 = NEW
RANDOMIZE USR 6137 = LIST (ab Zeile 0)
RANDOMIZE USR 7406 = STOP
RANDOMIZE USR 3438 = Löscht Zeilen 23 und 24

PRINT # 1;"xyz"	schreibt auf der Editierzeile
# 2;"xyz"	Schreibt auf dem Bildschirm oben
# 3;"xyz"	Schreibt auf dem Drucker
LIST # 3	listet auf dem Drucker
Input # 2;x	macht einen INPUT im oberen Bildschirmbereich

Speicherplätze: DISPLAY
:16384 — 22527
ATTRIBUTE
:22528 — 23297 = 768 Bytes (24*32)
Beispiel: POKE 22528,x :x=207 ergibt ein blinkendes helles gelbes Zeichen auf blauem Grund
x = INK * 8 + PAPER + 6 (BRIGHT) + 128 (FLASH)

Speicherplatz: Bildzähler
:23672 bis 23674 wird 50mal in der Sekunde um 2 erhöht
Beispiel: 10 POKE 23670,0: POKE 23673,0:POKE 23674,0
20 LET s=(PEEK 23673 +256*PEEK 23673) /502
25 PRINT s:REM s Zeit in Sec. seit Zeile 10
30 IF s >= 60 THEN GOTO 10

Großbuchstaben: POKE 23658,8 ein
0 aus

Wartezeit: 10 PRINT "Press any key to continue"
20 IF INKEYS$ = "" THEN GOTO 20
30 CLS: PRINT "This would be page 2"
Einfacher und vor allem Speicherplatz-sparender ist: PAUSE n n=0 bedeutet unendliche Pause; jeder Tastendruck setzt die Funktion zurück.

BREAK-Schutz: POKE 23659,0 (Sinclair User, Sept. 84,50/51) (dann kann man aber die untersten zwei Zeilen nicht im Programm (zum Beispiel INPUT) gebrauchten; bei SCROLL wird die Systemvariable DF SZ ((define screen size)) ebenfalls wieder auf 2 gesetzt, und der Schutz ist unwirksam; INKEY$ benutzen)

Fehlermeldungsvariable ERR SP verändern:
10 LET A= PEEK 23613+256*PEEK 23614:POKE A,b
Werte für b ausprobieren; zum Beispiel 100-150-200 oder 250. 200 = NEW
CP 3/84,93:
Folgende Zeilen einfügen:
1 LET a= PEEK 23613+256*PEEK 23614: POKE a,0:POKE a+1,0
9999 LET b= 23613:POKE b,0: POKE b+1,61: SAVE "xxx" CODE 23992, (PEEK 23641 + 256 * PEEK 23642) — 23552: RUN
Zum Abspeichern GOTO 9999, zum Laden LOAD "xxx" CODE eingeben. Damit wird schon ein Absturz verursacht, wenn man wäh-

rend des Ladevorgangs ein BREAK versucht.
Basic-Programmlänge: PRINT PEEK 23637+256*PEEK 23628-23755
9998 PRINT "Freier Speicherbereich=
"; (PEEK 23730+256*PEEK 23731-PEEK 23649-256*PEEK 23650) +229
9999 PRINT "Länge des Programms="; (PEEK 23641+256*PEEK 23642-PEEK 23635-256*PEEK 23636)-229

Folgende Zeile in ein Programm eingebaut listet x Zeilen:
10 POKE 23692,x+1: LIST: STOP (für x < 256)
POKE 23693,56 macht so manches Listing wieder sichtbar.
(K.-G. Scheffer)

LPrint III — Probleme gelöst

Das weit verbreitete Drucker-Interface LPrint III hat seine Tücken. Diese sind jedoch zu umgehen, wenn die Ursache erst einmal erkannt ist.

Beim Kauf meines Druckers Gemini-10X bestellte ich aus Bequemlichkeit ein LPrint III, was ich aber sehr bald bedauerte, als sich die in Ausgabe 3/85 der Happy-Computer beschriebenen Symptome zeigten. Zeichen gingen auf unerklärliche Weise verloren und bei gekauften Programmen verweigerte der Gemini zumeist seine Mitarbeit. Sofort kam mir der Verdacht, daß wohl mit dem Interface etwas sein mußte, was nicht der sehr dürftigen Anleitung zu entnehmen war, und ging daran, das Interface näher zu untersuchen. Äußerst hilfreich war dabei, die aus der Sicht des Herstellers vielleicht verständliche Methode, die Bezeichnung eines ICs abzuschleifen.

Erstes Ergebnis: Beim dem IC handelt es sich um einen einfachen Buffer vom Typ 74125 beziehungsweise 74425.

Ich zeichnete mir vom Layout einen Schaltplan und begann, die Funktion zu ergründen.
Betrachtet wird nur die Parallel-Schnittstelle.

Das Interface wird mit einem LPRINT-Kommando initialisiert.
Dabei geschieht nun folgendes:

Der Rechner läuft in die normale ZX-Printer-Routine bis zum Befehl IN A,(FB) bei Adresse OFOCH. Mit diesem Befehl wird das im LPrint III-Interface befindliche EPROM selektiert und das Spectrum-ROM ganz abgeschaltet. Der Einsprung ins ROM erfolgt bei Adresse 0712H=dezimal 1804. Dies ist durch die unvollständige Adreßcodierung des EPROMs (OFOCH-800H=0712H) möglich. Dann wird zuerst nachgesehen, ob schon eine Initialisierung stattgefunden hat, indem die Kanalinformation für die Druckroutine untersucht wird. Kritisch sind deshalb auch alle I/O-Einleseoperationen mit gesetztem A7 und zurückgesetztem A2, also alle IN-Befehle, die größer als 127 und nicht durch 4 teilbar sind. Bei all diesen Adressen besteht akute Absturzgefahr, weil immer das EPROM eingeschaltet wird. Man sollte dies beim Bau eigener Erweiterungen berücksichtigen. Nur eine vollständige Adreßdecodierung kann hier Abhilfe schaffen.

Adressen-Suche

Die Adresse der Kanalinformation für die Druckroutine findet man über die Systemvariable CHANS. Dort steht eine Adresse, zu der noch ein Offset von 15 addiert werden muß, um an die Adresse für die Druckroutine zu kommen.

In Basic: PRINT 256*PEEK 23632+PEEK 23631+15.

War das Interface noch nicht initialisiert, wird dorthin die Adresse OEFDH geschrieben und dient als Aufrufadresse für alle folgenden LPrint-Befehle, wodurch das EPROM in der vorher beschriebenen Weise immer angesprochen wird.

Mit jedem Aufruf werden nun bestimmte Maschinenprogrammteile aus dem EPROM in den Printer-Buffer kopiert. Dieser Kniff ist notwendig, um das EPROM wieder abschalten zu können und Zugriff auf verschiedene ROM-Routinen zu haben.

Um es nochmal zu betonen: Es wird kein einziges zu druckendes Zeichen im Printer-Buffer abgelegt, sondern ein Maschinenprogramm, das die eigentliche Aufgabe des Druckens übernimmt. Das Ausschalten des EPROMs erfolgt immer vom Printer-Buffer aus mit dem Befehl IN A,(7B).

Da das Maschinenprogramm mit jedem LPRINT-Befehl neu kopiert wird, spielt ein Überschreiben desselben eigentlich keine Rolle. Doch Vorsicht! Ein paar Speicherzellen im Printer-Buffer von Adresse 23543 bis 23548 werden als Zwischenspeicher benutzt. Werden diese überschrieben, so kann das unerwartete Folgen haben.

Suche nach dem Schuldigen

Dies ist einer der Gründe dafür, daß mit manchen gekauften Programmen ein Drucken nicht möglich ist, da sie genau diese Speicherzellen ebenfalls benützen.

Auch Tasword II macht da keine Ausnahme, wie die Befehle MOVE oder COPY BLOCK zeigen. (Deshalb auch manch unerklärliche Abstürze, wenn man irgendwann danach versucht, etwas zu drucken.) Erschwerend kommt hierbei noch dazu, daß zudem die Kanalinformation nach 23296 verzweigt wird, da von der Bedienungsanleitung ein "LPRINT CHR$ 5" beim Programmstart empfohlen wird. Damit wird das im Printer-Buffer stehende Maschinenprogramm immer direkt aufgerufen und nicht mehr ständig durch erneutes Einkopieren vom EPROM regeneriert.

Solange man nicht die Grafikzeichen als Druckersteuercode im Text verwendet, ist deshalb eine Initialisierung mit "LPRINT CHR$ 3" vorzuziehen.

Eine andere Möglichkeit, warum das Interface nicht mit einem gekauften Programm zusammenarbeitet, ist folgende:

Manche Programme (zum Beispiel Atric-Assembler) prüfen mit der Kombination IN A,(FB), BIT 6,A oder CP FF oder ähnliches nach, ob der ZX-Printer ansprechbar ist. Verläuft der Test negativ, so wird im normalen Programm ohne zu drucken

fortgefahren beziehungsweise wird ein "BUSY" des Printers angenommen.

Durch die Hardware der Interface bedingt ergibt ein derartiger Test immer (wegen der seriellen RS232-Schnittstelle) daß vermeintlich kein Drucker angeschlossen, beziehungsweise dieser nicht bereit ist. Gleichzeitig wird das ROM aus- und das EPROM eingeschaltet.

Hilfestellung mit Assembler

Abhilfe: Man durchsuche in Basic das Maschinencode-Programm nach dieser Bitkombination und ersetze sie durch die Kombination

```
IN A,($7B)
BIT 7,A
```

Beispiel:

```
10 FOR n=...TO....
20 IF PEEK n=219
AND PEEK (n+1)=
251 THEN GO SUB
100
30 NEXT n: STOP
100 IF (PEEK
(n+2)=203 AND
PEEK (n+2)=254 A
ND PEEK
(n+3)=255) THEN
PRINT "Änderung
bei Adresse: ";n:
POKE (n+1),123:
POKE (n+2),203:
POKE (n+3),127:
RETURN
110 PRINT "Initiali-
sierung bei Adres-
se: ";n
120 RETURN
```

Beim Artic-Assembler wird damit die Druckroutine lauffähig, indem nun "BUSY" an der richtigen Stelle (D7) abgefragt wird. Das Programm nennt einem gleichzeitig für andere Maschinencode-Programme die Adressen, bei denen das EPROM angesprochen wird und keine Änderung vorgenommen werden kann. In diesem Fall müssen die Adressen mit Hilfe eines Disassemblerprogramms untersucht und individuell abgeändert werden, was natürlich ein gewisses Maß an Erfahrung im Umgang mit Maschinensprache voraussetzt. Erfolgreich kann eine Änderung normalerweise vorgenommen werden, wenn irgendwo in der Nähe der genannten Adresse ein Aufruf mit:

```
LD A,2
CALL $1601    ;Kanal Printer öffnen
LD A,$0D
RST $10       ;Printroutine
```

zu finden ist. Der Befehl »IN A,(FB)« ist unbedingt auf »IN A,(7B)« abzuändern.

Unter Umständen muß man auch überprüfen, ob vielleicht die Adressen 23543 bis 23548 von dem Programm benutzt werden und diese abändern.

Ist das Interface einmal initialisiert, also die neue Kanalinfo eingeschrieben und der Printer-Buffer verändert, so läßt sich das Interface vollkommen neu initialisieren, indem man wieder die ursprüngliche Kanalinformation des ZX-Printers eingibt.

Neue Initialisierung des Interfaces

Vorteil: das Interface kann ohne Ausschalten des Rechners abgeschaltet und neu initialisiert werden.

```
POKE (256*PEEK 23632+PEEK 23631+15),244
POKE (256*PEEK 23632+PEEK 23631+16),9
```

Wer sich genauer für den Inhalt des EPROMs interessiert, der kann ihn sich auf folgende Art zugänglich machen, da über Basic kein Zugriff möglich ist.

```
LD DE,$9C40 ;Startadresse RAM
LD HL,0     ;Startadresse EPROM
LD BC,$800  ;2048 Speicherzellen kopieren
DI          ;Interrupt sperren
IN A,($FB)  ;EPROM einschalten
LDIR        ;Inhalt nach dez. 40000 kopieren
IN A,($7B)  ;EPROM ausschalten
EI          ;Interrupt freigeben
RET
```

Mann kann das Interface auch ohne die eingebaute Steuersoftware benutzen (ohne LPrint-Befehl), wie das nachstehende Basic-Programm zeigt:

```
10 INPUT a$
20 IF IN 123>=128 THEN GO TO 2
0: REM BUSY ?
30 FOF n=1 TO LEN a$
40 OUT 251,CODE a$(n): OUT
123,CODE a$(n): OUT 251,CODE
a$(n)
50 NEXT n
60 OUT 251,10: OUT 123,10: OUT
251,10: REM Vorschub
70 GO TO 10
```

Zum Schluß noch zu dem Problem, wenn scheinbar rein zufällig Zeichen verlorengehen:

Wenn man mit dem obigen Basic-Programm arbeitet, wird man feststellen, daß damit niemals ein Zeichen verlorengeht. So zeigt sich auch, daß es sich hierbei um einen Softwarefehler im EPROM handelt, der darin besteht, daß die BUSY-Leitung des Druckers nach der Übergabe eines Zeichens wieder zu früh abgefragt wird (noch bevor der Drucker sein Busy=1 setzen konnte). Also wird daraufhin ein Zeichen ausgegeben, obwohl der Drucker zu diesem Zeitpunkt noch nicht empfangen kann. Hier ist eine Umprogrammierung des EPROMs notwendig, bei der die Ausgabe eines Zeichens um ein paar Taktzyklen verzögert wird.

EPROM ändern

Stolze Besitzer eines EPROM-Programmiergerätes können nachstehende Änderung vornehmen, die den Fehler vollständig zum Verschwinden bringt.

Original dez.	Änderung
194 CALL $1F54	94 CALL $05BF
197 JP NC,$0D00	197 NOP
200 IN A,($7B)	198 LD A,0
	200 IN A,($7B)

Der Hersteller möge mir trotz Copyright den kurzen Auszug aus dem ohnehin fehlerhaften Programm verzeihen. Nach dieser Änderung wird der etwas längere BREAK-Test der LOAD-Routine aufgerufen.

Schaltungsbedingt muß ein EPROM 2716-1 (350ns) verwendet werden (auf keinen Fall eines mit 450ns).

Mit Hilfe der aufgezeigten Lösungswege sollte es einem versierten Happy-Computermann möglich sein, doch noch Freude an seinem LPrint III zu finden. (J. Holder)

Mit dem Spectrum auf Du und Du

Frei nach Murphys Gesetzen: Ein elektronisches Gerät wird beim Kauf getestet und funktioniert. Daheim verweigert es jedoch seinen Dienst. Nachahmung ohne jegliche Gewähr.

Im August 1984 hielt ich mich in Schottland auf. Als Besitzer eines Spectrums mußte ich die Gelegenheit für Einkäufe nutzen. Das Angebot in Großbritannien ist nicht nur viel reichhaltiger, sondern auch preiswerter als in Deutschland. Da auf meiner Wunschliste ganz oben das Interface 1 stand, habe ich mir dann in Edinburgh auch eines gekauft. 49,95 Britische Pfund wurden dafür bezahlt. Damit ich keine bösen Überraschungen erlebe, habe ich das Gerät im Laden ausprobiert. Es funktionierte.

Als ich wieder zu Hause war, wurde das Interface 1 angeschlossen, und dann gab es die große Überraschung: Es ging nicht. War es auf dem Transport kaputt gegangen? Am nächsten Tag habe ich es an einem anderen Spectrum auch ausprobiert. Dort funktionierte es. Also war mein Spectrum kaputt. Aber ohne Interface 1 lief er einwandfrei. Wie findet man den Fehler?

Ein paar Tage später war ich zufällig in Aachen. Mein Spectrum war von einer dortigen großen Versandfirma. Ich habe das Problem geschildert und die Auskunft war überraschend. Das Problem sei bekannt, denn das Interface 1 sei nicht kompatibel mit der Version II Spectrums. Es funktioniere nur mit Version III. Man wolle sich aber in England erkundigen, wie man das Problem lösen könne und ich solle in ein paar Wochen nochmals nachfragen. Ich kannte jedoch Leute, an deren Version II lief das Interface 1 einwandfrei, was mir zu denken gab.

Also selber rangehen hieß die Lösung. Ich habe einen Schaltplan besorgt. Und dann ging das Messen los. Um es kurz zu machen: nach etwa vier Stunden Arbeit haben wir den Fehler gefunden. Der Ausgang M1 meiner Z80A-CPU war defekt; und den braucht der normale Spectrum nicht, aber das Interface 1. Also den Z80A-Prozessor austauschen.

Aber vorher gab es noch ein weiteres Problem. Wir haben nämlich versucht den Spectrum provisorisch zu reparieren. Dazu wurde der Ausgang M1 über einen 4,7 kΩ Widerstand auf Null Volt gelegt und ein Kurzschluß gebaut. Ein Transistor gab seinen Geist auf. Computer reparieren ist also doch gefährlich! Das war ein ZTX 650 von Ferranti. Der war nur nirgends zu bekommen. Zum Glück gibt es einen Sinclair-Importeur in Ottobrunn. Nach einigen Telefonaten und drei Tage Warten hatte ich dann den Transistor. Also eingelötet und mein Spectrum ging wieder.

Erfolgserlebnis gesichert

Und die provisorische Lösung mit dem Widerstand hat auch funktioniert. Mein Spectrum stürzte lediglich von Zeit zu Zeit ab. Ergo doch die Z80A-CPU austauschen. Ich habe mir in Ottobrunn einen Kostenvoranschlag eingeholt. 85 Mark sollte Austausch und Sockeln der CPU kosten. Das ging ja noch. Aber wie lange wäre ich dann ohne Computer? Da blieb nur eines: Selbermachen.

Ich habe mir also einen IC-Sockel (3,50 Mark), eine neue CPU (14,80 Mark), eine Entlötpumpe (16,80 Mark) und eine feine Spitze für meinen Lötkolben (6,80 Mark) gekauft.

Nun ist das Auslöten eines 40poligen ICs nicht einfach. Alle Leute die berufsmäßig Löten, haben mir davon abgeraten. Aber mich schreckte nun nichts mehr.

Das Öffnen des Spectrums ist ja noch einfach. Man muß alle 40 Pins der CPU heiß machen und das flüssige Lötzinn absaugen. Aber vorsichtig. Macht man sie zu heiß, können sich die Leiterbahnen von der Platine ablösen. Und dann ist der Ofen aus. Nach vier vorsichtigen Durchgängen sahen alle Lötstellen sauber abgesaugt aus. Aber der IC war immer noch nicht los. Und mit Gewalt einfach ziehen ist auch gefährlich, da man dann leicht eine Leiterbahn abreißt. Aber es wurde doch noch geschafft. Man braucht nur mehr als zwei Hände dafür. (Zum Glück hat meine Frau mitgeholfen.) Man muß das Lötzinn an einem Beinchen erhitzen, und dann das Beinchen solange vorsichtig bewegen, bis das Lötzinn hart ist. Dann ist es lose. Hat man das vierzig mal gemacht, kann man versuchen, die CPU vorsichtig rauszuziehen. Fast 70 Minuten hatte es gedauert, bis die CPU draußen war.

Das Einlöten des Sockels war nun ein Kinderspiel. Beim Einstecken der neuen CPU muß man nochmals vorsichtig sein, damit man sich nicht elektrostatisch auflädt. Und dann kommt der große Moment. Geht er wieder oder geht er nicht. Bei mir ging er wieder. Und ich kann jetzt voller Stolz sagen: ich habe ihn selber repariert.

Rund 50 Mark habe ich investiert (mit dem kaputten Transistor) und die Entlötpumpe und die Lötspitze kann ich noch weiter gebrauchen. Also: Geld gespart und ein Erfolgserlebnis besonderer Art gehabt (aber auch geschwitzt). Jetzt zeigen sich auch keine Systemabstürze mehr, und die Fehlerrate bei den Microdrive-Operationen ist auch deutlich zurückgegangen. Und dann habe ich noch einiges über die Spectrum Hardware gelernt. Das ist sicher alles nichts Neues, aber wer hat schon alle Veröffentlichungen über den Spectrum gelesen.

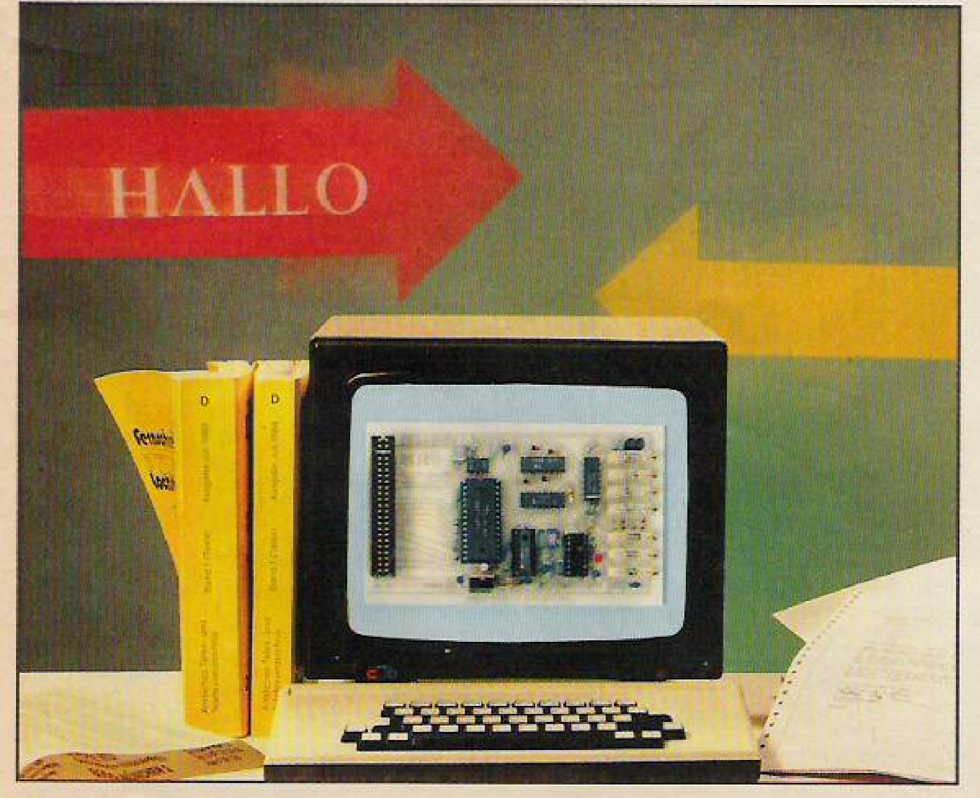
Der Anschluß —12 V Userport führt nicht diese Spannung, sondern eine hochfrequente Wechselspannung. Man kann allerdings mit einer Diode und einem Tantalelko daraus —12 V machen. Alle Videosignale am Userport (U, V, Y, Video) sind über Brücken geführt, die offensichtlich nicht in allen Spectrum eingelötet sind.

Bei einigen Spectrums ist der Platinenaufdruck für einige Bauteile (Elkos) falsch herum. Also aufpassen und nicht einfach anders herum einlöten. Und es gibt verschiedene Version II des Spectrum, sozusagen Version IIa und IIb. Ich besitze einen der Baureihe IIa. Man erkennt diesen daran, daß bereits alle sechs Logic-ICs zur Adressdecodierung im 16-KByte-Gerät eingelötet sind. Erst im IIb sind dafür Stackfassungen vorhanden. Auch die Brücke, mit der man die verschiedenen Typen der 32-KByte-Speichererweiterung unterscheidet, ist schon drin. Aber die kann man ja notfalls umlöten. Man muß also wirklich aufpassen. Sinclair ist immer für eine Überraschung gut (und wird es wohl bleiben). (R. W. Gerling)

Es hat gefunkt

Funksignale schwirren in Massen durch den Äther, aber nur wenige in Form von gesprochenen Worten, viele als Morsesignale und als Funkfernschreiben. Einige Signale sind sogar codierte Bildsignale und lassen sich per Programm decodieren.

Wer wäre wohl besser geeignet, diese »unverständlichen« Zeichen zu decodieren, als der Kollege Computer, zum Beispiel der Spectrum? Selbstverständlich darf nun nicht jeder jedes beliebige zu empfangende Signal auch »mitlesen«. Dafür gibt es schließlich reichlich Bestimmungen der deutschen Bundespost. Aber im Bereich des lizensierten Amateurfunks bleiben noch immer genug lesenswerte Funksignale über. Neben dem Tastfunk werden hier sowohl RTTY-Technik (radio-teletype) als auch die neueren Kommunikationsformen AMTOR und Packet-Radio genutzt, die eigentlich RTTY-Abwandlungen zum Betrieb automatischer Amateurfunk-Stationen (Mailboxen) sind, die ohne Computertechnik nicht denkbar sind. RTTY, also das Funkfernschreiben, wird normalerweise mit speziellen Fernschreibern codiert und decodiert. Mit einem entsprechenden Interface ausgerüstet, kann jedoch sogar der Spectrum sowohl zum Ausgeben als auch zum Empfangen dieser Signale verwendet werden. Das Interface ist hier im Monitor zu sehen. Es wird als Bausatz für 149 Mark oder als Fertigteil für 369 Mark angeboten. Das mitgelieferte Programm ist wohl der eigentliche Clou der Anlage. Es ist komplett in Z80-Maschinencode geschrieben und wird über die Tastatur bedient. Es prüft selbsttätig sowohl das Vorhandensein des Interfaces als auch den freien Speicherplatz und die angeschlossenen Microdrives. Anschließend wird der Benutzer zur Zeit- und Stationsnamen-Eingabe aufgefordert. Fünf Textspeicher mit je 253 Zeichen können beschrieben oder vom Recorder mit Text belegt werden. Laden und Speichern geht dabei recht schnell, da ohne Header gearbeitet wird. Das Programm verzweigt nun in zwei Richtungen: Einen Sende- und einen Empfangszweig. Text, der empfangen wird, kann so im Speicher abgelegt werden, daß er später entweder nochmals auf dem Bildschirm gezeigt wird oder aber ausgedruckt und/oder auf Microdrive abgespeichert wird. Der TX-(Sende-)Puffer kann 50000 Zeichen aufnehmen. Deren Aussendung wird von der CPU des Computers überwacht und die exakte Sendegeschwindigkeit (45 oder 50 BAUD) wird eingehalten. Am Schluß jeder Aussendung wird der Stationsname gesendet und der Spectrum schaltet wieder auf Empfang zurück.



Der Computer im Funkkreis

Es geht auch mit Bildsignalen

Wer nun meint, die Kräfte des kleinen Sinclair-Computers seien ausgereizt, der irrt sich gewaltig. Ohne jegliche Hardware-Zutaten ist der Spectrum in der Lage, sogenannte SSTV-Signale zu decodieren und auf dem Bildschirm auszugeben. SSTV steht für Amateur-Bildfunk und ist bisher wenigen Spezialisten vorbehalten. Vielleicht ändert sich aber die Zahl der Fernseh-Amateure durch den Computereinsatz schnell. Das entsprechende Spectrum-Programm benötigt für Sende- und Empfangsbetrieb den 48 KByte-Speicher. »Minicom SSTV« ist in Maschinensprache geschrieben und benötigt 10 KByte Programmspeicher. In dem RAM-Bereich von 25000 bis 50000 können bis zu sieben Bilder abgelegt werden, die im Laufe des Programms dann abgerufen werden. Hierbei werden Texte wie Bilder erfaßt und empfangen beziehungsweise gesendet. Für größere Empfangssicherheit sorgt ein aktives Filter, das man zwischen Empfänger und Spectrum schalten kann. Für die hier vorgestellten Anwendungen ist übrigens eine entsprechende Amateurfunk-Station samt Lizenz nötig.

Mit diesen beiden Spitzenprogrammen beweist der Spectrum eindeutig, daß er mehr kann als simple Morsetraineraufgaben wahrnehmen. Schade eigentlich, daß die reichlich vorhandene Fachliteratur diesen Kleincomputer bisher ausklammert. (mk)

Sound zum Selbermachen

Ein Sound-Generator-Bausatz verwandelt die mageren »BEEPs« zu vollem Klang und läßt sich auch noch zur 8-Bit-PIO erweitern. Es ist ein »update« der in Happy-Computer 11/84 vorgestellten Version.

Ausgangspunkt für diese Bauanleitung ist der Sound-IC AY 3-8912 von General Instruments. Dieser IC ist (richtig angewendet) ein wahres Klangwunder, denn er enthält drei Tongeneratoren, einen Rauschgenerator, drei Hüllkurvengeneratoren, ein Steuerregister und, um das Maß voll zu machen, noch eine 8-Bit-PIO (siehe Bild).

»OUT« macht's möglich

Angesprochen werden die 14 Register des AY 3-8912 durch den Befehl »OUT 189,n« (n = 0 bis 14 für Register 0 bis 14). Verdeutlicht wird dies durch die Register-Tabelle. Mit »OUT 191,n« wird der gewünschte Wert n in das vorher angewählte Register eingeschrieben.

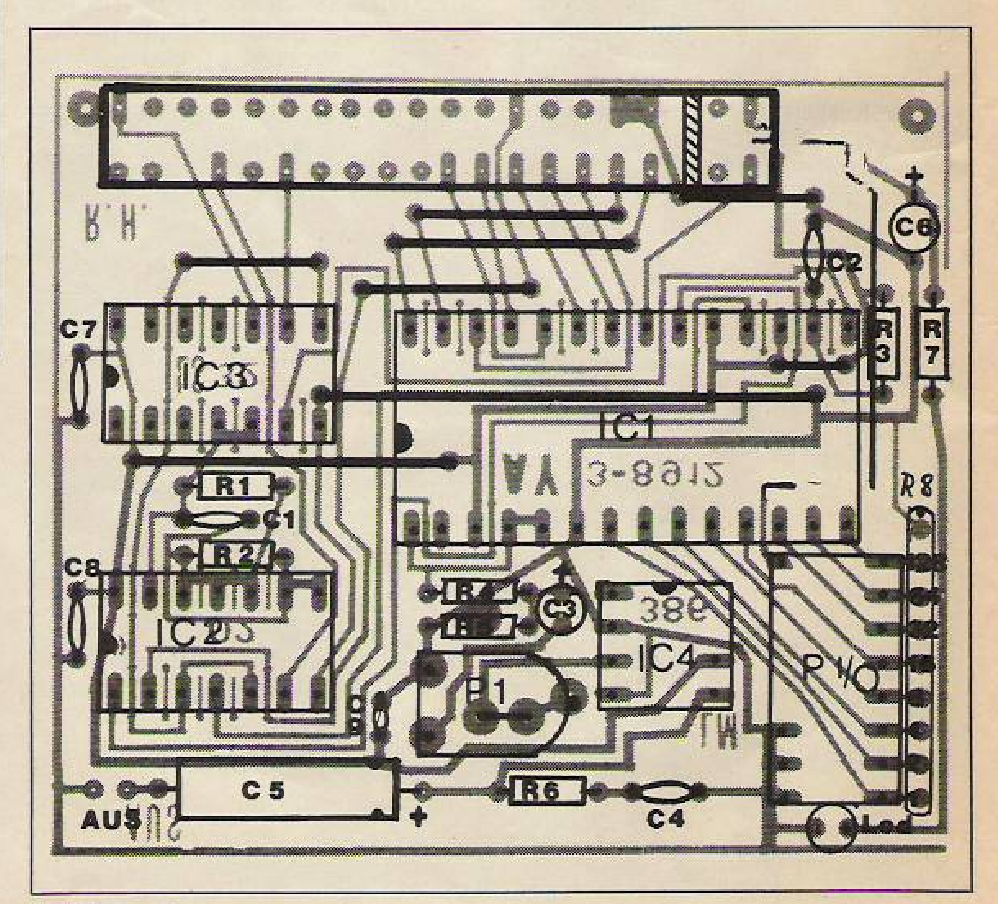
Die OUT-Befehle 189 und 191 werden durch die Gatter a3, a4 und b2 aus dem IORQ und dem WR-Signal sowie aus den Adreßleitungen A1 und A6 an der Steckerleiste des Spectrum decodiert.

Als letztes bleibt noch der NF-Verstärker, aufgebaut aus dem IC LM 386 und ein paar diskreten Bauteilen. Bei diesem IC handelt es sich um einen 0,5 W-Verstärker für niedrige Betriebsspannungen (in unserem Fall 5 V).

Sound- und PIO-Baustein ▶



◀ Platine des Sound-Moduls



Bestückungsplan

Funktionsgruppen des Sound-Chips AY 3-8912:

1. Der Rauschgenerator, dessen Klang in 32 Stufen (hell bis dunkel) verändert werden kann.

2. Die drei Tongeneratoren, die jeweils 4095 Töne erzeugen, einstellbar durch ein 12-Bit-Wort (8 Bit fein, 4 Bit grob).

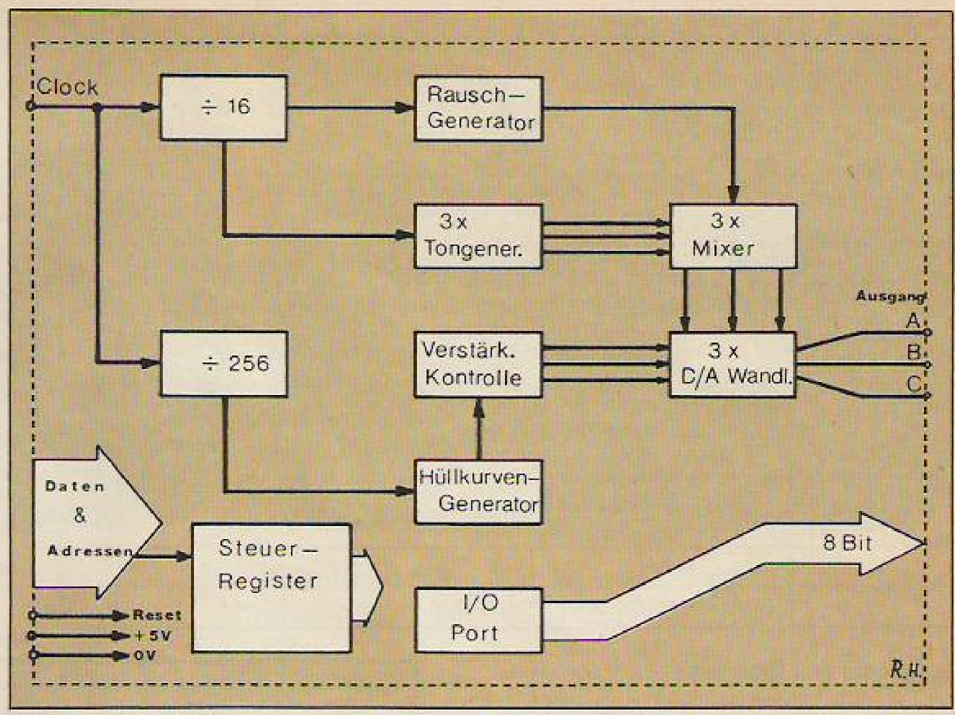
3. Das Steuerregister als wichtigster Block. Durch dieses Register wird festgelegt, welcher Klang am Ausgang erscheint und ob die PI/O auf Ein- oder Ausgabe gestellt ist. Einstellbar ist dieses Register durch eine Zahl von 0 bis 255.

4. Die steuerbaren Verstärkerstufen, mit denen die Lautstärke am Ausgang für jeden einzelnen Kanal festgelegt wird (15 Stufen). Gibt man hier eine 16 ein, so wird der betreffende Kanal auf den Hüllkurvengenerator geschaltet.

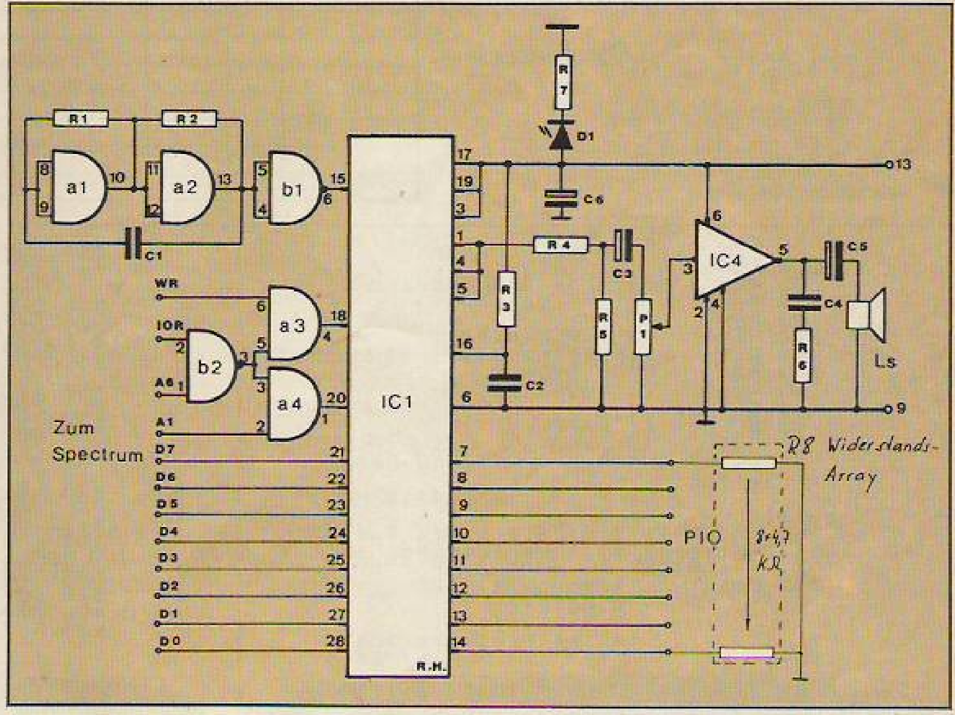
5. Der Hüllkurvengenerator, eingestellt durch Register 11 und 12. Mit dieser Stufe wird die Hüllkurvenfrequenz ausgewählt. Das geschieht für beide Register zusammen durch ein 16-Bit-Wort, so daß bei der angegebenen Dimensionierung des Clock-Generators, bestehend aus IC 74LS32, Gatter a1 und a2, eine Frequenz von 0,1 Hz bis zirka 8 kHz eingestellt werden kann.

6. Das Steuerregister für den Hüllkurvengenerator.

7. Der parallele Input/Output-Port (PIO), angesprochen durch Register 14, Einstellung mit 1 bis 255 für den Output-Port. Wird in Register 7 eine 1 in Datenbit 6 eingegeben (also die Zahl 64), so ist die PI/O auf Ausgabe eingestellt. Eine 0 in Register 7 stellt die PI/O auf Eingabe. Mit »PRINT IN 191« kann der Eingangswert aufgelistet werden.



Das Innenleben des Sound-Chips



Das Schaltbild für mehr Sound

Stückliste:

Widerstände:		Kondensatoren:		Halbleiter:	
R1,R2	470Ω	C1	330pF	IC 1	AY 3-8912
R3	10KΩ	C2	0,1µF	IC 2	74LS02
R4	220Ω	C3	4,7µF	IC 3	74LS32
R5	1KΩ	C4	47nF	IC 4	LM 386
R6	10Ω	C5	100µF	D 1	Led 3mm
R7	150Ω	C6	10µF		
P1 Poti	4,7KΩ	C7,C8,C9	10nF		
R8 Widerstandsarray 8x4,7kΩ					

Zubehör:

IC-Fassungen 2x14 pol.
1x28 pol.
1x 8 pol. (nur bei Bedarf)
1x16 pol.
Sinclair-Stecker
Lautsprecher 0,2 bis 0,5W/8 Ohm
Klinkenbuchse 3mm

Stückliste zum Soundgenerator

Die Hardware der Musikbox ist einfach, aber wirkungsvoll

Der Aufbau auf der abgebildeten Platine sollte dem einigermaßen geübten Bastler keine Schwierigkeiten bereiten. Es ist jedoch unbedingt auf saubere Lötstellen, richtige Po-

Register–Nr.	Funktion			
0	unteres Byte von	0	255	Frequenz Kanal A
1	oberes Byte von	0	15	
2	unteres Byte von	0	255	Frequenz Kanal B
3	oberes Byte von	0	15	
4	unteres Byte von	0	255	Frequenz Kanal C
5	oberes Byte von	0	15	
6	Rauschgenerator	0	31	Klang des Rauschgenerators
7	Steuerregister: eine1 im Bit 0,1,2 sperrt Tonkanal A,B,C eine1 im Bit 3,4,5 sperrt Rauschkanal A,B,C			
8	Lautstarke Kanal A	0	15	bei 16 wird der Kanal auf den Hüllkurvengenerator geschaltet
9	Lautstarke Kanal B	0	15	
10	Lautstarke Kanal C	0	15	
11	unteres Byte von	0	255	Frequenz Hüllkurve
12	oberes Byte von	0	255	
13	Hüllkurvenform	0	15	
14	Parallel–Input/Output–Port (PI/O)			

▲ **Register und Funktionen**

lung der Tantalkondensatoren und die Vollzähligkeit der Drahtbrücken zu achten, um den Spectrum nicht in die ewigen Jagdgründe zu schicken.

Bei dem Verstärker-IC werden die Beinchen 7 und 8 nicht gebraucht und können abgekniffen werden. Die diversen Drahtbrücken auf der Platine sind unbedingt als erstes einzulöten, da sie teilweise unter den ICs liegen. Man kommt später nämlich nicht mehr ran.

Etwas schwieriger ist schon der Einbau der bestückten Platine in das hier vorgeschlagene Gehäuse. Der Kreativität des Einzelnen sind hier keine Grenzen gesetzt. Platine, Bausatz und ein Musikprogramm sind beim Autor erhältlich.

(R. Hobmeier)

Signalformen ▶

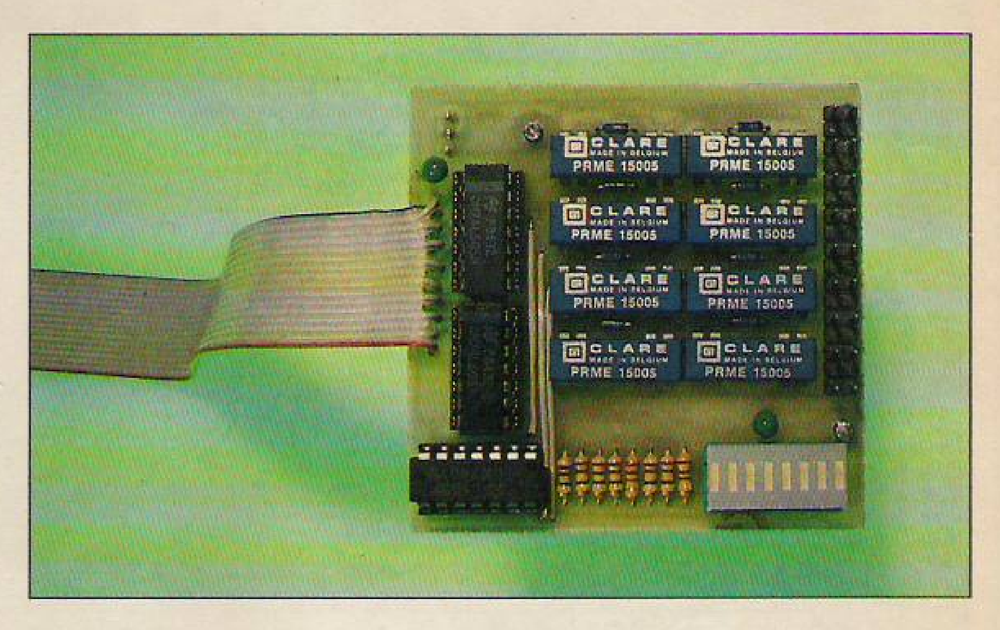
Dezimal	Abfall	Anstieg	Vibrato	Halten	Kurvenform
0	0	0	X	X	
4	0	I	X	X	
8	I	0	0	0	
9	I	0	0	I	
10	I	0	I	0	
11	I	0	I	I	
12	I	I	0	0	
13	I	I	0	I	
14	I	I	I	0	
15	I	I	I	I	

Demo-Programm
▼ **zum Soundgenerator**

```
  10 REM          Demo-Programm
  20 REM               fuer
  30 REM    Spectrum-Sound-Board
  35 REM               von
  36 REM
  40 REM     Ralf Hobmeier
  41 REM     Schlenkhoffsweg 27
  42 REM     4720 BECKUM
  43 REM     Tel.02521/4570
  44 REM
  45 BORDER 4: PAPER 6
  50 CLS
  51 PRINT INK 2;"       Sound-Dem
o-Programm
von                         Ralf H
obmeier"
  55 PRINT AT 5,0;"Sie haben fol
gende Klaenge zur  Verfuegung:":
 PRINT : PRINT
  60 PRINT "   1.Sound-Effekt
               2.Rakete
               3.Explosion
               4.Testprogramm (Be
ispiel in                      Ze
ile 401)"
  65 RESTORE : PRINT INK 3;AT 15
,0;"Bitte Ziffer eingeben."
  70 INPUT "Welchen Klang wuensc
hen Sie?";a$
  75 IF a$<"1" OR a$>"5" THEN PR
INT AT 20,0;"Falsche Eingabe!":
PAUSE 50: PRINT AT 20,0;"
                   ": GO TO 65
  80 IF a$="1" THEN GO TO 100
  85 IF a$="2" THEN GO TO 200
  90 IF a$="3" THEN GO TO 300
  95 IF a$="4" THEN GO TO 400
 100 REM [illegible]
 105 CLS : PRINT OVER 1; INK 2;A
T 10,5;"SOUND-EFFEKT"
 110 LET a=189: LET b=191
 120 OUT a,7: OUT b,248
 130 OUT a,8: OUT b,15
 140 OUT a,0
 150 FOR n=0 TO 128: OUT b,(n-64
*INT (n/64))
 160 OUT b,128-n: OUT b,n: NEXT
n: OUT b,0: PAUSE 50
 170 FOR m=1 TO 10: FOR n=0 TO 2
55 STEP 10: OUT b,n: NEXT n: NEX
T m
 180 GO TO 500
 200 REM [illegible]
 205 CLS : PRINT OVER 1; INK 2;A
T 10,5;"RAKETE"
 210 LET a=189: LET b=191
 220 OUT a,7: OUT b,254: FOR n=0
 TO 255: OUT a,8: OUT b,INT (n/1
6)
 230 OUT a,0: OUT b,n: NEXT n
 240 DATA 6,13,7,7,8,16,9,16,10,
16,12,50,13,3
 250 FOR m=1 TO 7: READ c: READ
d: OUT a,c: OUT b,d: NEXT m
 255 PAUSE 100
 260 GO TO 500
 300 REM [illegible]
 305 CLS : PRINT OVER 1; INK 2;A
T 10,5;"EXPLOSION"
 310 LET a=189: LET b=191
 320 DATA 6,31,7,7,8,16,9,16,10,
16,12,20,13,3
 330 FOR n=1 TO 7: READ e: READ
f: OUT a,e: OUT b,f: NEXT n
 335 PAUSE 100
 340 GO TO 500
 400 REM [illegible]
 401 REM Werte zum testen:
          0     =50
          1     =0
          2     =55
          3     =0
          4     =60
          5     =0
          6     =0
          7     =248
          8     =16
          9     =16
         10     =16
         11     =0
         12     =50
         13     =3
 405 LET a=189: LET b=191
 406 CLS
 410 INPUT "Register?";r
 420 INPUT "Daten?";d
 430 OUT a,r: OUT b,d
 440 IF r>16 THEN GO TO 490
 445 IF r=15 THEN GO TO 50
 450 FOR n=0 TO 14: OUT a,n: PRI
NT AT (n+1),3;n: PRINT AT (n+1),
10;IN a;"  ": NEXT n
 455 PRINT AT 20,0;"Zurueck zum
Menue mit >15<"
 465 GO TO 410
 490 FOR m=0 TO 14: OUT 189,m: O
UT 191,0: NEXT m
 495 GO TO 400
 500 FOR n=0 TO 14: OUT 189,n: O
UT 191,0: NEXT n
 510 GO TO 45
```

PIO zum Soundgenerator

Nachdem das bereits in »Happy-Computer 11/84« vorgestellte Sound-Modul hoffentlich von allen Nachbauern zum Klingen gebracht werden konnte, wollen wir uns nun mit der eingebauten PIO beschäftigen. Der Ausbau ist relativ einfach.



In der nachfolgenden Bauanleitung wird sie nur als Ausgabe-Port mit 8 Relais-Ausgängen benutzt. Die Relais-Kontakte können mit maximal 500 mA belastet werden.

Wie wir uns noch erinnern können, war an dem Sound-Generator eine 16polige IC-Fassung als Ausgang vorgesehen. Von hier aus geht es nun mit einem Flachbandkabel mit 16poligem IC-Stecker zur PI/O-Platine.

Bevor wir aber mit dem Aufbau beginnen, müssen wir noch eine wichtige Änderung am Sound-Generator vornehmen, damit die 5 V-Stromversorgung des Spectrum nicht überlastet wird. Diese Änderung ist in dem vorstehenden Sound-Artikel bereits eingearbeitet.

Die auf dem Bild gekennzeichnete Leiterbahn durchtrennen und das freie Beinchen der IC-Fassung mit +9 V an der Steckerleiste verbinden. Die PI/O-Platine hat einen eigenen 5 V-Regler.

Die Funktionsweise ist folgende:

Um den Sound-Chip auf Daten-Ausgabe vorzubereiten, sind einige Eingaben nötig.

Als erstes alle Register des PSG auf »0« stellen.

FOR I = 0 TO 14 : OUT 189,I: OUT 191,0 : NEXT I

Dann die PI/O initialisieren:

OUT 189,7 : OUT 191,64 (Für Fachleute: Das 6. Datenbit im Register 7 muß für PI/O-Ausgabe auf »H« und für PI/O-Eingabe auf »L« gesetzt werden.)

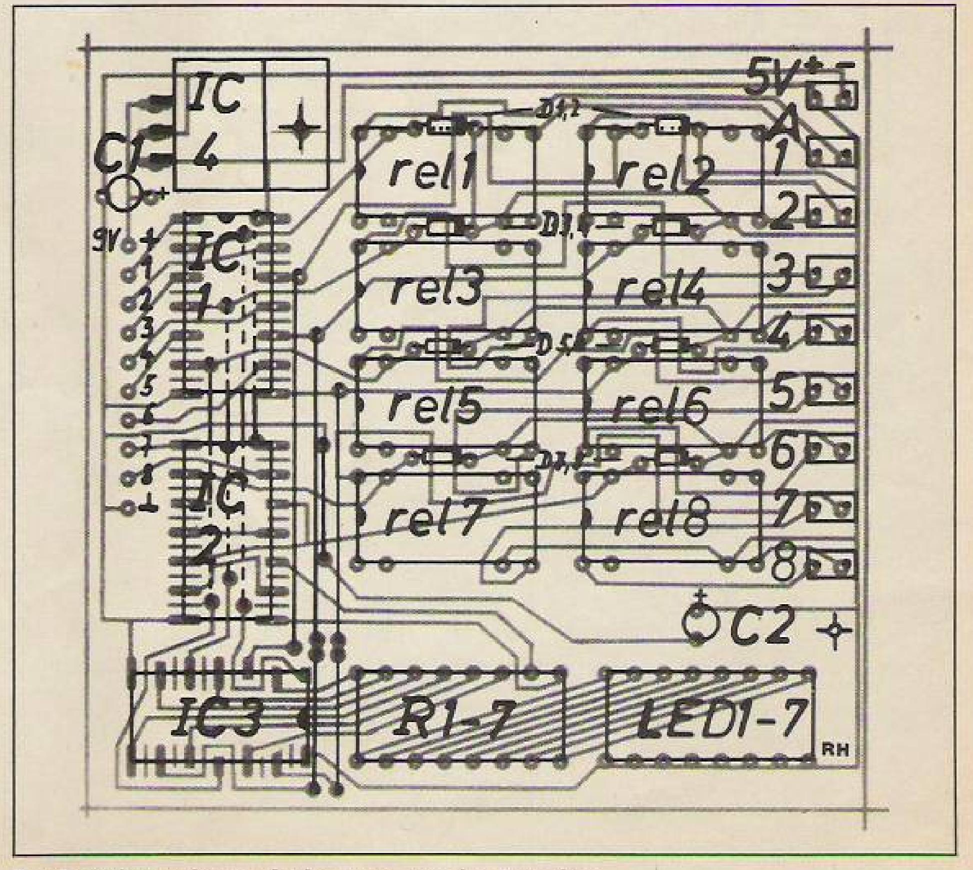
Nun können wir mit »OUT 189,14 : OUT 191,n« jeden Ausgang von 1-8 einschalten.

Da die PI/O aber binär codiert ist, müssen wir für die einzelnen Ausgabe-Befehle (n) für den

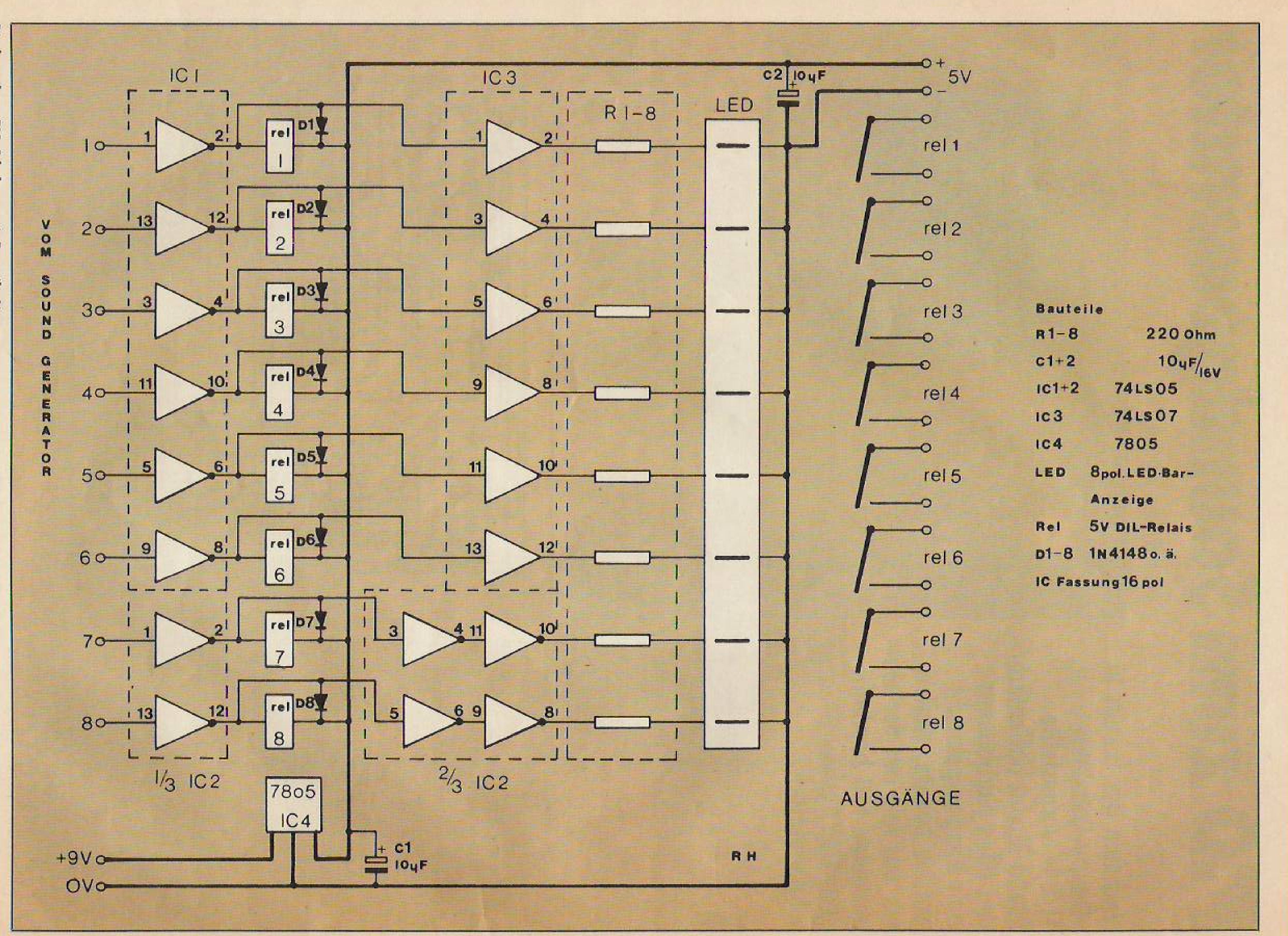
1. Ausgang eine 1
2. Ausgang eine 2
3. Ausgang eine 4
4. Ausgang eine 8
5. Ausgang eine 16
6. Ausgang eine 32
7. Ausgang eine 64
8. Ausgang eine 128 eingeben.

Beispiel:

Um den 7. Ausgang einzuschalten, muß der Befehl »OUT 189,14 : OUT 191,64« heißen. Der Relais-Kontakt ist



Der Bestückungsplan in gleicher Lage, wie das Foto oben.



Die komplette PIO-Schaltung mit Bauteile-Liste

jetzt geschlossen. Das Ausschalten geschieht durch das Setzen des Registers 14 auf »0« oder mit dem Einschalten eines anderen Ausganges. Sollen mehrere Ausgänge gleichzeitig eingeschaltet werden, so sind die Binär-Zahlen zu addieren, zum Beispiel für Ausgang »1« und Ausgang »7« einschalten muß die Zahl 65 eingegeben werden (1 + 64).

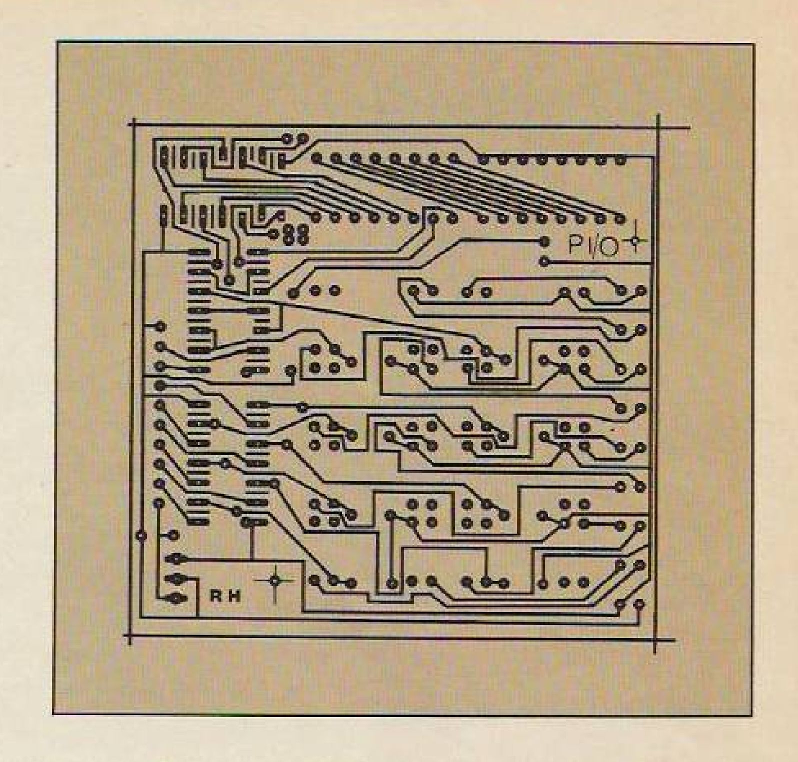
Die ICs 1-3 dienen dem PSG-IC AY3-8912 als Treiberstufen, da der AY nur eine LS-Last treiben kann.

Mit der LED-Balken-Anzeige kann der eingeschaltete Ausgang sichtbar gemacht werden. Als Relais werden 5 V DIL-Relais verwendet. Die Platine wird nach dem Bestückungsplan aufgebaut und verlötet. Die ICs werden ohne Fassung eingelötet.

Der Spannungsregler wird auf der Rückseite der Platine mit der Aufdruckseite nach unten auf einem Kühlblech montiert, das ungefähr die Größe der Platine hat. Diese Konstruktion kann dann mit Abstandsbolzen in den vorgesehenen Löchern verschraubt werden. Bei dem Anschluß des Flachbandkabels empfiehlt sich größte Sorgfalt, da es IC-Stecker mit unterschiedlicher Beschaltung gibt. Hier sollte man zur Vorsicht alle Kontakte mit einem Ohm-Meter durchmessen. Platine, Bauteile und auch ein kompletter Bausatz können beim Autor bezogen werden.

Nun viel Spaß und Erfolg.

(R. Hobmeier)



Blick auf die Leiterbahn-Seite der Platine

```
   10 REM ************************
          *                      *
          * PI/O-STEUERPROGRAMM  *
          *         VON          *
          *    RALF HOBMEIER     *
          *                      *
          ************************
   11 BORDER 6: PAPER 6: INK 0: C
LS
   12 PRINT AT 21,27;"R.H."
   13 PLOT 0,30: DRAW 160,0: DRAW
 0,27: DRAW -160,0: DRAW 0,-27
   15 REM ALLE REGISTER AUF 0
   20 LET A=189: LET B=191
   30 FOR N=0 TO 14: OUT A,N: OUT
 B,0: NEXT N
   35 REM PI/O AUF AUSGABE
   40 OUT A,7: OUT B,64
   45 GO SUB 100
   46 PRINT INK 1;AT 8,8;"0 1 2 3
 4 5 6 7 8"
   49 PRINT AT 15,1;"PI/O-AUSGABE
 ";AT 16,5;"ZUM        =";AT 17,1;
" REGISTER 14": LET C$="0": GO T
O 56
   50 PRINT AT 0,0;"WELCHER AUSGA
NG SOLL ANGESTEUERTWERDEN ?  BIT
TE EINGEBEN         WEITERE FUNKT
IONEN "; FLASH 1;">9<": PAUSE 4E
4: LET C$=INKEY$
   51 IF C$<"0" OR C$>"9" THEN GO
 TO 50
   53 FOR n=8 TO 24 STEP 2: PRINT
 AT 10,n;" ": NEXT n
   55 REM PI/O  WERT AUSGEBEN
   56 LET D=VAL C$: PRINT INK 1;A
T 8,10;"1 2 3 4 5 6 7 8 ": PRINT
 INK 2;AT 10,VAL C$+D+8;"█"
   60 IF C$="3" THEN LET D=4
   65 IF C$="4" THEN LET D=8
   70 IF C$="5" THEN LET D=16
   75 IF C$="6" THEN LET D=32
   80 IF C$="7" THEN LET D=64
   85 IF C$="8" THEN LET D=128
   86 IF C$="9" THEN GO TO 150
   90 OUT A,14: OUT B,D
   95 PRINT INK 1;AT 16,15;D;"  "
   96 GO TO 50
  100 PLOT 77,85: DRAW 126,0: PLO
T 77,98: DRAW 126,0
  110 FOR I=85 TO 98: PLOT 76,I:
PLOT 91,I: PLOT 107,I: PLOT 124,
I: PLOT 140,I: PLOT 156,I: PLOT
172,I: PLOT 188,I: PLOT 204,I: N
EXT I
  120 RETURN
  145 REM LAUFLICHT
  150 BORDER 6: PAPER 6: INK 0: C
LS
  155 OUT 189,7: OUT 191,64
  160 PRINT "LAUFLICHT";"   TEMPO
 AENDERN MIT"; FLASH 1;">T<"
  170 PRINT AT 21,27;"R.H."
  180 PLOT 0,30: DRAW 160,0: DRAW
 0,27: DRAW -160,0: DRAW 0,-27
  190 PRINT AT 15,1;"PI/O-AUSGABE
 ";AT 16,5;"ZUM        =";AT 17,1;
" REGISTER 14"
  200 PLOT 77,85: DRAW 126,0: PLO
T 77,98: DRAW 126,0
  210 FOR I=85 TO 98: PLOT 76,I:
PLOT 91,I: PLOT 107,I: PLOT 124,
I: PLOT 140,I: PLOT 156,I: PLOT
172,I: PLOT 188,I: PLOT 204,I: N
EXT I
  215 INPUT "PAUSENLAENGE ?";C
  220 LET A=1: LET B=1: LET C$="1
"
  225 LET D$=INKEY$: IF D$="T" OR
 D$="t" THEN GO TO 215
  230 PRINT AT 10,A+9;"█"
  240 PRINT AT 10,A+9;"█";AT 10,A
+7;" "
  245 OUT 189,14: OUT 191,B
  246 PRINT AT 16,15;B;"  ": LET
B=B+B
  248 PAUSE C
  249 LET A=A+2
  250 IF A=17 THEN PRINT AT 10,A+
7;" ": GO TO 220
  280 GO TO 240
```

Basic-Testlisting zur PIO

Markt&Technik-Buchverlag

Depot-Händler

Tragen Sie Ihre Buchbestellung und die Anschrift des Depotbuchhändlers auf die Bestellkarte in diesem Heft ein. Bitte vergessen Sie den Absender nicht.

Buchhandlung Herder, Kurfürstendamm 69
1000 Berlin 15, Tel. (5030) 8835002,
BTX *921782#
Computare Fachbuchhandlung, Keithstraße 18
1000 Berlin 30, Tel. (030) 2139021
Thalia Buchhaus, Große Bleichen 19
2000 Hamburg 36, Tel. (040) 3005050
Boysen + Maasch, Hermannstraße 31
2000 Hamburg 1, Tel. (040) 30050515
Electro-Data, Wilhelm-Heidsiek-Straße 1
2190 Cuxhaven, Tel. (04721) 51288
Buchhandlung Muehlau, Holtenauer Straße 116
2300 Kiel, Tel. (0431) 85085
ECL, Norderstraße 94-96
2390 Flensburg, Tel. (0461) 28181
Buchhandlung Weiland, Königstraße 79
2400 Lübeck, Tel. (0451) 74006-09
Buchhandlung Storm, Langenstraße 10
2800 Bremen 1, Tel. (0421) 321523
Buchhandlung Lohse-Eissing, Marktstraße 38
2940 Wilhelmshaven, Tel. (04421)41687
Buchhandlung Schmorl u. v. Seefeld, Bahnhofstraße 13
3000 Hannover 1, Tel. (0511) 327651
Buchhandlung Graff, Neue Straße 23
3300 Braunschweig, Tel. (0531) 49271
Deuerlich'sche Buchhandlung, Weender Straße 33
3400 Göttingen, Tel. (0551) 56808
Buchhandlung an der Hochschule, Holländische Straße 22
3500 Kassel, Tel. (0561) 83807
Stern Verlag, Friedrichstraße 24-26
4000 Düsseldorf, Tel. (0211) 373033
Buchhandlung Baedeker, Kettwiger Straße 33-35
4300 Essen 1, Tel. (0201) 221381
Regensberg'sche Buchhandlung, Alter Steinweg 1
4400 Münster, Tel. (0251) 40541-5
Buchhandlung Acker, Johannisstraße 51
4500 Osnabrück, Tel. (0541) 28488
Buchhandlung Lensing, Westenhellweg 86-88
4600 Dortmund, Tel. (0231) 16980
Buchhandlung Brockmeyer, Querenburger Höhe 281/Unicenter
4630 Bochum, Tel. (0234) 701360
Buchhandlung Meier + Weber, Warburger Straße 98
4790 Paderborn, Tel. (05251) 63172
Buchhandlung Phönix GmbH, Obernstorwall 25
4800 Bielefeld 1, Tel. (0521) 69071
Buchhandlung Gonski, Neumarkt 24
5000 Köln 1, Tel. (0221) 210528
Mayer'sche Buchhandlung, Ursulinerstraße 17-19
5100 Aachen, Tel. (0241) 48142
Buchhandlung Behrendt, Am Hof 5a
5300 Bonn 1, Tel. (0228) 658021
Buchhandlung Cusanus, Schloßstraße 12
5400 Koblenz, Tel. (0261) 36239
Akad. Buchhandlung Interbook, Fleischstraße 61-65
5500 Trier, Tel. (0651) 43596
Buchhandlung W. Finke, Kipdorf 32
5600 Wuppertal 1, Tel. (0202) 454220
Buchhandlung Balogh, Sandstraße 1
5900 Siegen, Tel. (0271) 55298-9
Buchhandlung Naacher, Steinweg 3
6000 Frankfurt 1, Tel. (069) 298050
Buchhandlung Wellnitz, Lautenschlägerstraße 4
6100 Darmstadt, Tel. (06151) 76548
Buchhandlung Feller + Gecks, Friedrichstraße 31
6200 Wiesbaden, Tel. (06121) 304911
Ferber'sche UNI-Buchhandlung, Seltersweg 83
6300 Gießen, Tel. (0641) 12001
Sozialwissenschaftliche Fachbuchhandlung, Friedrichstraße 24
6400 Fulda, Tel. (0661) 75077
Gutenberg Buchhandlung, Große Bleiche 29
6500 Mainz, Tel. (06131) 37011
Buchhandlung Bock + Seip, Futterstraße 2
6600 Saarbrücken, Tel. (0681) 30677
Buchhandlung Wilhelm Hofmann, Bismarckstraße 98
6700 Ludwigshafen, Tel. (0621) 516001
Buchhandlung Loeffler, B 1,5
6800 Mannheim 1, Tel. (0621) 28912
Buchhandlung Stehn, Bahnhofstraße 13
7000 Stuttgart 50, Tel. (0711) 561476
Buchhandlung am Markt, Kramstraße 6
7100 Heilbronn, Tel. (07131) 68682
PCB Micro-Computer, Oskar-Kalbfell-Platz 8
7410 Reutlingen, Tel. (07121) 270443
UNI Buchhandlung Kellner + Moessner, Kaiserstraße 18
7500 Karlsruhe, Tel. (0721) 691436
Buchhandlung Roth, Hauptstraße 45
7600 Offenburg, Tel. (0781) 22097
Rombach Center, Bertholdstraße 10
7800 Freiburg, Tel. (0761) 49091
Fachbuchhandlung Hofmann, Hirschstraße 4
7900 Ulm, Tel. (0731) 60949
Schauties Elektronik, Bachstraße 52
7980 Ravensburg, Tel. (0751) 26138
Buchhandlung Hugendubel, Marienplatz
8000 München 2, Tel. (089) 2389-1
Computerbücher am Obelisk, Barerstraße 32-34
8000 München 2, Tel. (089) 282383
Pele's Computerbücher, Schillerstraße 17
8000 München 2, Tel. (089) 781752
Universitätsbuchhandlung Lachner, Theresienstraße 43
8000 München 2, Tel. (089) 521340
Buchhandlung Schönhuber, Theresienstraße 6
8070 Ingolstadt, Tel. (0841) 33146/47
Computerstudio Gertrud Friedrich, Ludwigstraße 3
8220 Traunstein, Tel. (0861) 14767
Buchhandlung Pustet, Kl. Exerzierplatz 4
8390 Passau, Tel. (0851) 56945
Buchhandlung Pustet, Gesandtenstraße 6
8400 Regensburg, Tel. (0941)53061
Buchhandlung Dr. Büttner, Adlerstraße 10-12
8500 Nürnberg, Tel. (0911) 232318
STS Computer Vertrieb, Werner-Siemens-Straße 19
8580 Bayreuth, Tel. (0921) 62320
Burger Elektro, Leimitzer Straße 11-13
8670 Hof, Tel. (09281) 40075
Sortiments- u. Bahnhofsbuchh. J. Strykowski, Bahnhofplatz 4
8700 Würzburg, Tel. (0931) 54389
Buchhandlung Pustet, Grottenau 4
8900 Augsburg, Tel. (0821) 35437
Kemptener Fachsortiment, Salzstraße 30
8960 Kempten, Tel. (0831) 14413

Belgien:
Eicher Micro & Personal Computer, Hünningen 56-58
B-4780 St. Vith, Tel. (080) 227393

Luxemburg:
Librairie Promoculture, 14, rue Duchscher (Pl. de Paris)
L-1011 Luxembourg-Gare, Tel. 480691, Telex 3112

Schweiz:
Buchhandlung Meissner, Bahnhofstraße 41
5000 Aarau, Tel. (064) 247151
Bücher Balmer, Neugasse 12
6300 Zug, Tel. (042) 214141
Buchhandlung Enge, Bleicherweg 56
8002 Zürich, Tel. (01) 2012078
Buchhandlung Orell Füssli, Pelikanstraße 10
8022 Zürich, Tel. (01) 2118011
Freihofer AG, Wissenschaftliche Buchhandlung, Universitätsstr. 11
8033 Zürich, Tel. (01) 3634282
Buchhandlung am Rößlitor, Webergasse 5
9001 St.Gallen, Tel. (071) 228726

Autorenverzeichnis

Aicher, Richard
Redaktion Happy-Computer

Baran, Bernhard
Postfach 35
6710 Frankenthal

Bosetti, Peter
Rott 53
6294 ML Vejlen
Holland

Brandl, Horst
Schumacherring 27
8000 München 83

Gerling, R.W.
Hollergasse 16
8551 Heroldsbach

Gleissner, Simon
Bingenheimerstr. 46
6361 Reichelsheim

Henrich, Axel
Heidhoernweg 21
2962 Spetzerfahn

Hilpert, Martin
Kopernikusweg 4
8630 Coburg

Hönnig, Michael
Postfach 1123
2980 Norden 2

Holder, Jürgen
Meisenweg 13
7319 Dettingen/Teck

Hobmeier, Ralf
Schlenkhoffsweg 27
4720 Beckum

Krawczyk, Karina
Redaktion Happy-Computer

Leuhardt, Heinrich
Redaktion Happy-Computer

Malik, Markus
Schlesischer Weg 4
4775 Lippetal

Pallada, H.
J. Jelgerhuishof 6
1065 TM Amsterdam

Prillinger, Horst
Jahnstraße 2
A-5280 Braunau am Inn

Roth, Uwe
Irscherstraße 15
5500 Trier

Scheffer, K.-G.
Michaelstr. 20
5169 Heimbach 4

Schimice, Axel
Fontanestraße 8
5620 Velbert 1

Schober, Klaus
Ansbacher Straße 54
8820 Gunzenhausen

Voelkers, Oliver
Brombeerweg 5
4500 Osnabrück

Wilhelm, Harald
Förster Str. 16
5000 Köln 30

v. Zeschwitz, Hans Georg
Schwindstr. 17
2000 Hamburg 52

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)
Stellv. Chefredakteur: Michael Scharfenberger (sc)
Redakteur: mk = Manfred Kotting (177)
Redaktionsassistenz: Monika Lewandowski (222)
Fotografie: Jens Jancke

Layout: Leo Eder (Ltg.), Alexander Gerhardt

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug,
Tel. 042-223155/56, Telex: 862329 mut ch
USA: M&T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. 415- 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkauf: Brigitta Fiebig (211)

Anzeigenverwaltung und Disposition: Patricia Schiede (172)

Vertriebsleitung, Werbung: Hans Hörl (114)

Verlagsleiter M&T Buchverlag: Günther Frank (212)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 764830

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-238. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Bezugspreis: Das Einzelheft kostet DM 14,-.

Druck: St. Otto-Verlag GmbH, Laubanger 23, 8600 Bamberg

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly (120)
Für Anzeigen: Ralph Peter Rauchfuß (126).

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 5-22052

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München, Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr

Neu: Pelikan Disketten. Die richtige Qualität für Sie.

Jetzt gibt es vom Spezialisten für Computer-Zubehör auch Disketten. In Pelikan Qualität:

Premium Class
100% fehlerfrei und
absolut zuverlässig
auch unter extremsten Bedingungen

im praktischen 2er-Pack oder 10er-Vorrats-Pack. In den Größen 5¼" und 3½"*, einseitig oder doppelseitig und in einfacher, doppelter oder »vierfacher« 96 tpi-Dichte. Alle Qualitäts-Disketten mit Verstärkungsring.

Zur sicheren und geschützten Aufbewahrung Ihrer Pelikan-Disketten: Die Pelikan Disketten-Box in 2 Größen (für 40 oder 80 Disketten). Staubdicht und abschließbar. Und wie praktisch: Disketten und Zubehör sowie Kassetten für Ihren PC gibt es jetzt aus einer Hand in Ihrem Bürobedarfs-Fachgeschäft.



premium class
100% fehlerfrei
absolut zuverlässig, auch unter extremsten Bedingungen

Pelikan macht die Arbeit sichtbar leichter.

Pelikan

*) 3½"-Disketten ab Anfang 1985